SU-SEL-67-099

A Real-Time Digital Spectrum Analyzer

by

R. B. McCullough

November 1967

Scientific Report No. 23

Prepared under
National Aeronautics and Space Administration
Research Grant No. NsG-377

RADIOSCIENCE LABORATORY

STANFORD ELECTRONICS LABORATORIES

STANFORD UNIVERSITY • STANFORD, CALIFORNIA

SEL-67-099

A REAL-TIME DIGITAL SPECTRUM ANALYZER

by

R. B. McCullough

November 1967

Scientific Report No. 23

Prepared under

National Aeronautics and Space Administration
Research Grant No. NsG-377

Radioscience Laboratory
Stanford Electronics Laboratories
Stanford University Stanford, California

Sci. Rept. 23
NsG 377
December 1967

ABSTRACT

In the past, spectral analysis has been done almost exclusively by analog equipment. The main stumbling block for digital spectral analysis has been the enormous number of computations necessary to calculate the spectrum digitally.

This research develops a new algorithm for digital spectral analysis in real time. This new algorithm is intended for implementation by special-purpose circuitry using currently available integrated circuits. Maximum sampling rates above 1 MHz can be realized in continuous real-time operation. This is much faster than any other existing algorithm can operate.

A closed-form analytical expression is developed for the passband characteristics of the discrete Fourier transform operation. This expression is used to evaluate in detail the effectiveness of several time-domain and frequency-domain operations aimed at improving the passband characteristics.

A hardware design is presented which makes use of certain novel features of this new real-time algorithm. Among these features is the use of shift registers in which only the inputs and outputs are available. This permits use of very long integrated circuit shift registers in very small packages since very few inputs and outputs are required. This feature would not be an advantage in realizing this algorithm in a general-purpose computer, but it definitely is in special-purpose hardware.

PRECEDING PAGE BLANK NOT FILMED.

CONTENTS

TABLE

ILLUSTRATIONS

ACKNOWLEDGMENT

The guidance and encouragement of Dr. Allen Peterson and Dr. Bruce Lusignan during the progress of this research are gratefully acknowledged.

Chapter I

INTRODUCTION

Spectrum analysis has historically been performed almost exclusively by analog equipment. Digital equipment has not been used because of the tremendous number of computations that must be performed in order to calculate the spectrum.

In 1965 Cooley and Tukey derived a new algorithm which reduced the number of computations necessary by the factor of $(\log_2 n)/n$ for calculating n spectrum points from n input samples. This algorithm and its offspring have made practicable the computation of spectra on large general-purpose computers. These applications have generally been either non-real time or real time with very low frequency components.

A new algorithm is derived here which is suitable for real-time operation at sample rates exceeding 1 MHz. This algorithm is intended for implementation by special-purpose hardware using integrated circuits. It also features continuous rather than "batch" processing.

Considerable attention is paid to the passband characteristics of these algorithms. A convenient closed-form analytic expression is derived for the passband of the basic Fourier transform algorithm. This expression is used to evaluate in detail the effectiveness of several time-domain and frequency-domain operations aimed at improving the passband characteristics.

A hardware design is presented which uses currently available integrated circuits. The maximum sampling rate obtainable is determined to be above 1 MHz. The author feels that the algorithm and design presented offer the best solution for rapid real-time spectral analysis.

Chapter II

SPECTRUM ANALYSIS

A. Definition of Spectra

Spectrum as used here will refer to either the voltage spectrum, the energy spectrum, or the power spectrum of the input signal. The voltage spectrum is the Fourier transform of the input signal as given by

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-i\omega t}\, dt \qquad (2.1)$$

The voltage spectrum will in general be complex.

The energy spectrum is obtained simply by taking the magnitude squared of the voltage spectrum:

$$E(\omega) = |F(\omega)|^2 = \left| \int_{-\infty}^{\infty} f(t)\, e^{-i\omega t}\, dt \right|^2 \qquad (2.2)$$

The energy spectrum is purely real.

A problem arises in the use of the energy spectrum since real signals of unbounded time duration possess energy spectra which are unbounded. In general, the energy per unit time, i.e., the power, will be bounded. The power spectrum can be defined as

$$P(\omega) = \lim_{T\to\infty} \frac{1}{2T} \left| \int_{-T}^{T} f(t)\, e^{-i\omega t}\, dt \right|^2 \qquad (2.3)$$

The power spectrum $P(\omega)$ is also purely real and in addition is bounded for most input signals.

An alternate way of arriving at the power spectrum is to take the autocorrelation function of the input and then Fourier transform it:

$$P(\omega) = \int_{-\infty}^{\infty} \left[\lim_{T\to\infty} \frac{1}{2T} \int_{-T}^{T} f(\tau)\, f(t+\tau)\, d\tau \right] e^{-i\omega t}\, dt \quad (2.4)$$

It is apparent that no matter which spectrum is desired or what path is followed to obtain it, some form of Fourier transform will be involved.

B. Analog Spectrum Analysis

Spectrum analysis has traditionally been done with analog equipment. Two basic systems have been used: filter banks and swept analyzers. The most straightforward approach is shown in Fig. 1. In this system a bank of n tuned band-pass filters is used. Each filter passes components of the input signal that lie within a narrow band of frequencies between $\omega_j - \Delta\omega/2$ and $\omega_j + \Delta\omega/2$, where ω_j is the center frequency of the filter and $\Delta\omega$ is its bandwidth. The output of each filter is then squared and averaged to give an estimate of the signal power at the frequency ω_j. The circuitry is repeated n times to give n power spectrum estimates.

The second approach is illustrated in Fig. 2. In this system the input signal is mixed with the output of a swept-frequency oscillator to produce the sum and difference frequencies. The difference frequency is then passed through a

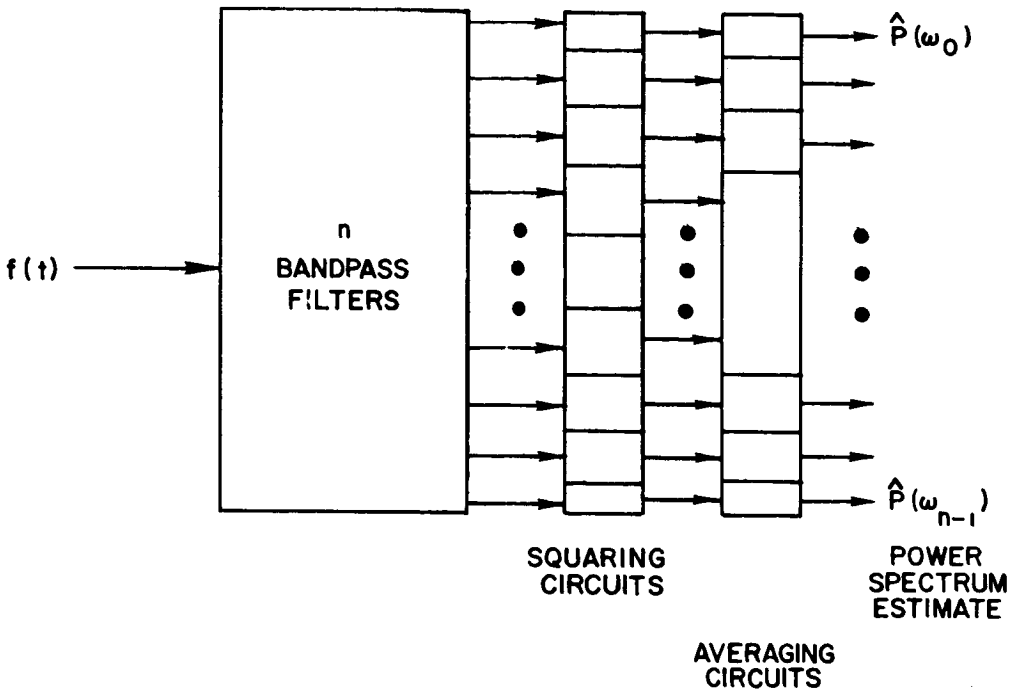


Fig. 1. ANALOG FILTER BANK ANALYZER.

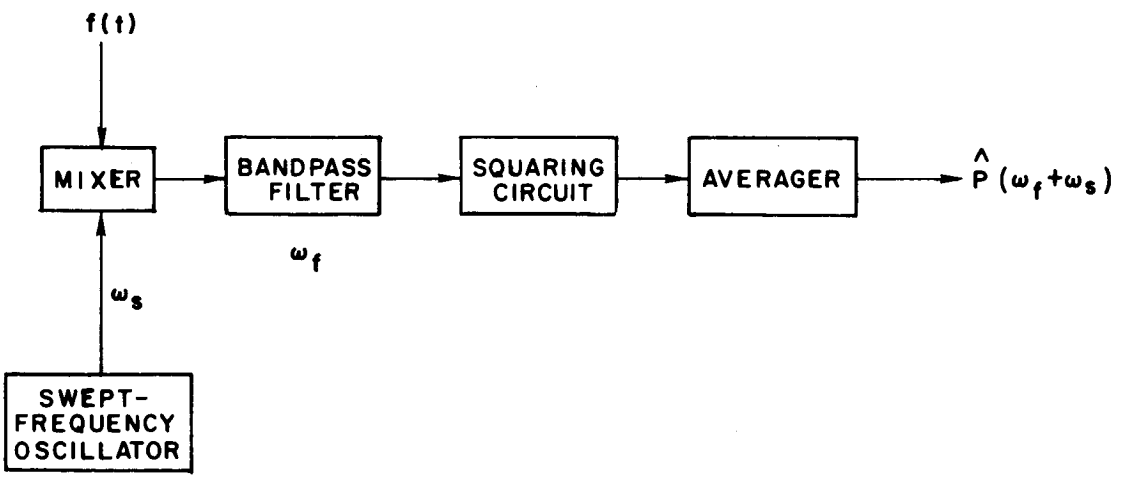


Fig. 2. SWEPT-FREQUENCY ANALYZER.

bandpass filter, squared, and averaged to produce an estimate of the power spectrum at a frequency $\omega_s + \omega_f$, where ω_s is the oscillator frequency and ω_f is the filter frequency. As the oscillator frequency is varied, the frequency at which the power spectrum is estimated varies. This analyzer requires considerably less hardware than the filter-bank analyzer.

C. Digital Spectrum Analysis

Digital spectrum analysis may be performed in a number of ways. In all cases, the analog input signal is sampled and converted into a series of digital numbers. Calculations must then be performed on these numbers to produce another series of numbers which represent the spectrum of the signal being analyzed. The problem lies in being clever enough in the organization of the calculations that they may be performed in a minimal amount of time by a minimal amount of hardware.

Weaver, Mantey, Lawrence, and Cole at Stanford University [1966] have used a set of linear difference equations to obtain an estimate of the spectrum of the input. The analyzer they implemented is shown in Fig. 3. Each of the difference equations is of the form:

$$y(nT) = a_0x(nT) + a_1x[(n-1)T] + \ldots + a_kx[(n-k)T] - b_1y[(n-1)T] - b_2y[(n-2)T] - \ldots - b_my[(n-m)T] \quad (2.5)$$

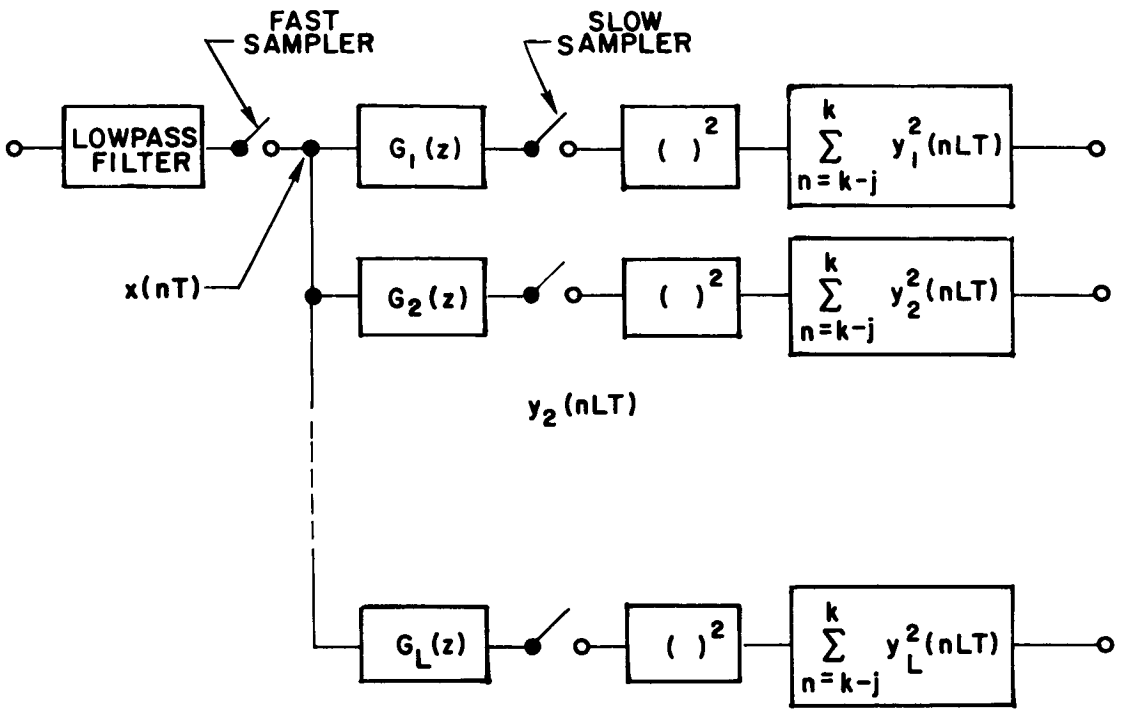


Fig. 3. DIGITAL SPECTRUM ANALYZER USING LINEAR DIFFERENCE EQUATIONS.

Since the previous outputs are included as inputs to the analyzer, long time spans of data can be included in relatively few calculations to yield good resolution in the frequency domain. Unfortunately the inclusion of previous outputs also introduces the possibility of instability into the difference equations. This analyzer is a direct digital equivalent of the analog filter-bank analyzer.

Other digital spectrum analyzers have been built by digitally computing a Fourier transform of the input samples. By and large these analyzers have been realized on general-purpose digital computers and have been operated in non-real time. The transform algorithms used have been some form of

the Cooley-Tukey algorithm or a derivation thereof. These algorithms "batch-process" the data; that is, computation is not begun until a batch of n samples is collected. The processing produces n outputs and then halts until the next set of n inputs is collected. The batch-process nature of these algorithms makes them unsuited for continuous real-time operation.

The present effort also utilizes a digital Fourier transform. However, a new algorithm is derived which permits efficient real-time operation. The block diagram of the proposed analyzer is shown in Fig. 4. The analog input signal f(t) is sampled and converted into a digital number; these numbers are used in the computation of the Fourier transform of the input. Finally, the magnitude of the transform is squared to obtain the power spectrum of the input. The particular algorithm used is especially well suited to realization by special-purpose hardware built from integrated circuits.

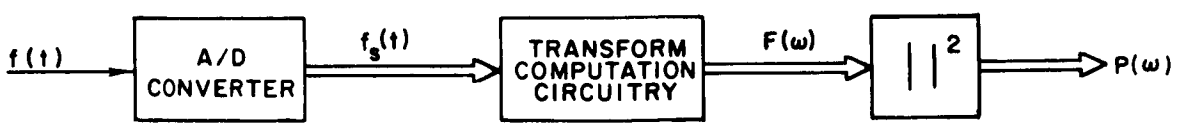


Fig. 4. REAL-TIME DIGITAL SPECTRUM ANALYZER.

D. Comparison of Analog and Digital Analyzers

The analog analyzer has historically held the edge in practical implementation because it has required far less

hardware than the digital spectrum analyzer. The analog analyzers have suffered from problems characteristic of any analog instrument. In particular, it is difficult to keep the gain and frequency bands of a large number of analog filters from drifting with time, temperature, and other environmental changes. On the other hand, the characteristics of the digital analyzer are completely insensitive to its environment.

The major stumbling block to the implementation of digital spectrum analyzers has been the large number of computations necessary to perform the digital Fourier transform. This problem has been alleviated greatly by the Cooley-Tukey algorithm which is derived in detail in Chapter III. Furthermore, the advent of integrated circuits has made the building of large, special-purpose computational units a practicable approach. The digital spectrum analyzer proposed in this research will become even more attractive as integrated circuit technology progresses.

Chapter III

DISCRETE FOURIER TRANSFORMS

A. Discrete Fourier Transform Equations

The basic operation in any spectrum analysis is the taking of the Fourier transform. In our digital spectrum analyzer we will be working with samples of the input taken every τ seconds. The transform we compute will in fact be the transform of the sampled input rather than the true input.

The sampling can be represented mathematically as multiplication of the input $f(t)$ by an impulse train of period τ. The resultant $f_s(t)$ is given by

$$f_s(t) = f(t) \sum_{k=-\infty}^{\infty} \delta(t - k\tau) \tag{3.1}$$

The Fourier transform of the sampled signal is given by

$$F_s(\omega) = \int_{-\infty}^{\infty} f(t) \sum_{k=-\infty}^{\infty} \delta(t - k\tau)\, e^{-i\omega t}\, dt \tag{3.2}$$

Carrying out the integral yields

$$F_s(\omega) = \sum_{k=-\infty}^{\infty} f(k\tau)\, e^{-i\omega k\tau} \tag{3.3}$$

Equation (3.3) is the basic equation for the digital computation of spectra. It involves taking the samples of the input $f(k\tau)$, multiplying by a complex number $e^{-i\omega k\tau}$,

and summing over all the available samples. These operations must be performed for every desired value of ω. Consequently, if we want n points of the spectrum and we have n time samples to work from, we will have to perform n^2 complex multiplications and additions.

B. Efficient Operation Grouping

If the operations are grouped appropriately we can make some savings in the number of operations performed. Let us look at the problem of calculating n samples of $F_s(\omega)$ evenly spaced between $\omega = 0$ and $\omega = [(n-1)2\pi]/n\tau$ from the n input samples of $f_s(t)$ from $t = 0$ to $t = (n-1)\tau$. Equation (3.3) may be written

$$F_s\left(\omega = \frac{j2\pi}{n\tau}\right) = \sum_{k=0}^{n-1} f(k\tau)\, \exp\left(-i\, \frac{j2\pi}{n\tau}\, k\tau\right) \qquad (3.4)$$

where the equation is to be evaluated for all values of j from $j = 0$ to $j = n-1$.

Let us write the indices j and k as binary numbers

$$j = j_{m-1}2^{m-1} + j_{m-2}2^{m-2} + \ldots + j_1 2 + j_0 \qquad (3.5)$$

$$k = k_{m-1}2^{m-1} + k_{m-2}2^{m-2} + \ldots + k_1 2 + k_0 \qquad (3.6)$$

By writing j and k as above we have made the tacit assumption that the number of samples is less than or equal to 2^m. Let us now shorten our notation in the following manner:

$$F_s\left(\omega = \frac{j2\pi}{n\tau}\right) = F_s(j_{m-1}, j_{m-2}, \ldots, j_1, j_0) = F_s(j) \quad (3.7)$$

$$f(t = k\tau) = f(k_{m-1}, k_{m-2}, \ldots, k_1, k_0) = f(k) \quad (3.8)$$

With this shortened notation, Eq. (3.4) can be written

$$F_s(j) = \sum_{k=0}^{n-1} f(k) \exp\left[-i \frac{2\pi}{n} \left(j_{m-1}2^{m-1} + \ldots + j_0\right)\right.$$
$$\left. \cdot \left(k_{m-1}2^{m-1} + \ldots + k_0\right)\right] \quad (3.9)$$

The product in the exponential contains powers of 2 between 2^0 and 2^{2m-2}. Let us write this product out in more detail:

$$2^{2m-2} j_{m-1}k_{m-1} + 2^{2m-3} \sum_{r=m-2}^{m-1} j_{2m-3-r}k_r + \ldots$$
$$+ 2^{m+1} \sum_{r=2}^{m-1} j_{m+1-r}k_r + 2^m \sum_{r=1}^{m-1} j_{m-r}k_r$$
$$+ 2^{m-1} \sum_{r=0}^{m-1} j_{m-1-r}k_r + 2^{m-2} \sum_{r=0}^{m-2} j_{m-2-r}k_r + \ldots$$
$$+ 2^2 \sum_{r=0}^{2} j_{2-r}k_r + 2 \sum_{r=0}^{1} j_{1-r}k_r + j_0k_0 \quad (3.10)$$

Notice that this product is in the form of powers of 2 with integer coefficients.

The maximum number of samples we have provided for is 2^m. Let us then take full advantage of our capability by setting the number of samples $n = 2^m$. Doing this, and using (3.10), we can expand the exponential of (3.9) as follows:

$$\exp\left(-i\,\frac{2\pi}{n}\,jk\right) = \exp\left[-i\,\frac{2\pi}{n}\,\left(j_{m-1}2^{m-1} + \ldots + j_0\right)\left(k_{m-1}2^{m-1} + \ldots + k_0\right)\right]$$

$$= \exp\left(-i2\pi 2^{m-2} j_{m-1}k_{m-1}\right)\, \exp\left(-i2\pi 2^{m-3} \sum_{r=m-2}^{m-1} j_{2m-3-r}k_r\right)\ldots$$

$$\cdot\, \exp\left(-i2\pi 2 \sum_{r=2}^{m-1} j_{m+1-r}k_r\right)\, \exp\left(-i2\pi \sum_{r=1}^{m-1} j_{m-r}k_r\right)$$

$$\cdot\, \exp\left(-i2\pi 2^{-1} \sum_{r=0}^{m-1} j_{m-1-r}k_r\right)\, \ldots\, \exp\left(-i2\pi 2^{2-m} \sum_{r=0}^{2} j_{2-r}k_r\right)$$

$$\cdot\, \exp\left(-i2\pi 2^{1-m} \sum_{r=0}^{1} j_{1-r}k_r\right)\, \exp\left(-i2\pi 2^{-m} j_0k_0\right) \qquad (3.11)$$

The first m-1 exponentials in (3.11) are of the form $\exp(-i2\pi p)$, where p is an integer, and hence are identically equal to one. Equation (3.11) may consequently be greatly simplified to:

$$\exp\left(-i\,\frac{2\pi}{n}\,jk\right) = \exp\left(-i2\pi 2^{-1}\sum_{r=0}^{m-1} j_{m-1-r}k_r\right)\exp\left(-i2\pi 2^{-2}\sum_{r=0}^{m-2} j_{m-2-r}k_r\right)\cdots$$

$$\cdot\exp\left(-i2\pi 2^{2-m}\sum_{r=0}^{2} j_{2-r}k_r\right)\exp\left(-i2\pi 2^{1-m}\sum_{r=0}^{1} j_{1-r}k_r\right)$$

$$\cdot\exp\left(-i2\pi 2^{-m}j_0k_0\right) \qquad (3.12)$$

In Eq. (3.12), k_{m-1} appears only in the first exponential; k_{m-2} only in the first two; k_{m-3} only in the first three; etc. Regrouping the exponentials, we have

$$\exp\left(-i\,\frac{2\pi}{n}\,jk\right) = \exp\left(-i2\pi k_{m-1}j_0 2^{-1}\right)\exp\left[-i2\pi k_{m-2}\left(j_1 2^{-1} + j_0 2^{-2}\right)\right]$$

$$\cdot\exp\left[-i2\pi k_{m-3}\left(j_2 2^{-1} + j_1 2^{-2} + j_0 2^{-3}\right)\right]\cdots$$

$$\cdot\exp\left[-i2\pi k_1\left(j_{m-2}2^{-1} + \cdots + j_0 2^{-m+1}\right)\right]$$

$$\cdot\exp\left[-i2\pi k_0\left(j_{m-1}2^{-1} + \cdots + j_0 2^{-m}\right)\right] \qquad (3.13)$$

The expression for the exponential in (3.13) may be put back into the summation in (3.9) and the summation over k may be broken into many sums over the individual k_r. The result is

$$F_s(j) = \sum_{k_0}\left\{\sum_{k_1}\left\{\cdots\left\{\sum_{k_{m-2}}\left\{\sum_{k_{m-1}} f(k)\exp\left(-i2\pi k_{m-1}j_0 2^{-1}\right)\right\}\right.\right.\right.$$

$$\left.\left.\left.\cdot\exp\left[-i2\pi k_{m-2}\left(j_1 2^{-1} + j_0 2^{-2}\right)\right]\right\}\cdots\right\}$$

$$\cdot \exp\left[-i2\pi k_1\left(j_{m-2}2^{-1} + \ldots + j_0 2^{-m+1}\right)\right]\Bigg\}$$

$$\cdot \exp\left[-i2\pi k_0\left(j_{m-1}2^{-1} + \ldots + j_0 2^{-m}\right)\right] \qquad (3.14)$$

Each of the variables k_r being summed upon in Eq. (3.14) has only the value 0 or 1, since each is a coefficient of a power of 2 in the binary number representation of k.

Equation (3.14) may be evaluated by beginning at the innermost sum and working outward. Let us arrange the 2^m samples of $f(k)$ in a column in order of ascending values of the binary number k. These samples are shown in the left-hand column of Fig. 5 for $m = 4$, or 16 samples. Figure 5 is a flow graph for the transform computations.

The innermost sum of (3.14) is over $k_{m-1} = 0, 1$ and can be written out as

$$\sum_{k_{m-1}=0}^{1} f(k) \exp\left(-i2\pi k_{m-1} j_0 2^{-1}\right)$$

$$= f(0,k_{m-2},\ldots,k_0)\, e^0 + f(1,k_{m-2},\ldots,k_0) \exp\left(-i2\pi j_0 2^{-1}\right) \qquad (3.15)$$

Thus the inner sum is seen to be a function of the k_{m-2}, $k_{m-3},\ldots,k_0$ and j_0. There will be 2^m of these functions which we will label $F(j_0,k_{m-2},\ldots,k_0)$. These are shown in the second column of Fig. 5 arranged in order of ascending

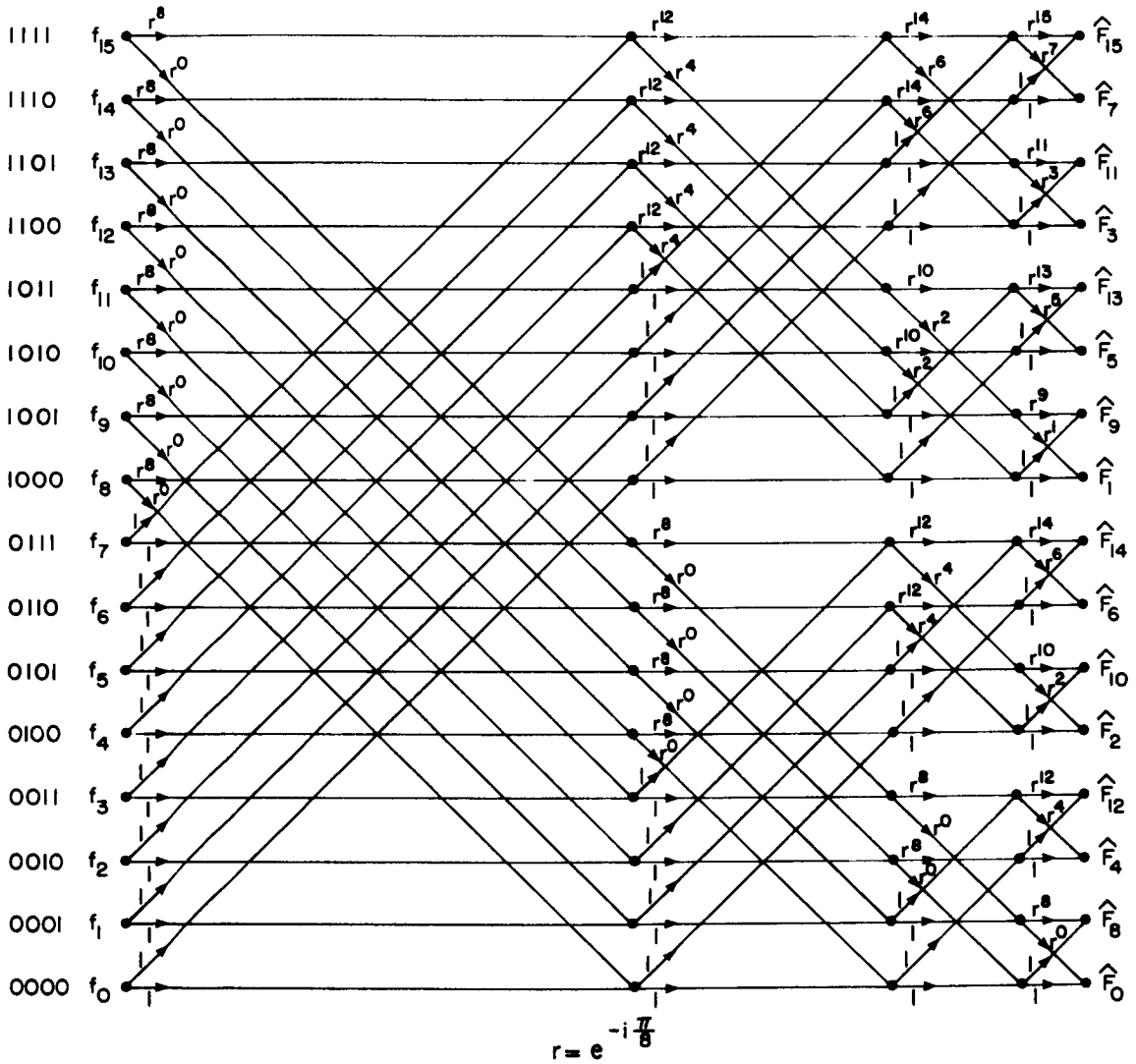


Fig. 5. FLOW CHART OF CALCULATIONS FOR $n = 2^4$.

value of the binary numbers $j_0 2^{m-1} + k_{m-2} 2^{m-2} + \ldots + k_0 2^0$. The nodes represent the value of the functions and are arrived at by summing the value of all incoming arrows multiplied by the value of their respective origins.

The next innermost sum of (3.14) can be expanded as

$$\sum_{k_{m-2}=0}^{1} F(j_0, k_{m-2}, \ldots, k_0) \exp\left[-i2\pi k_{m-2}\left(j_1 2^{-1} + j_0 2^{-2}\right)\right]$$

$$= F(j_0, 0, k_{m-3}, \ldots, k_0)\, e^0 + F(j_0, 1, k_{m-3}, \ldots, k_0)$$

$$\cdot \exp\left[-i2\pi\left(j_1 2^{-1} + j_0 2^{-2}\right)\right] \qquad (3.16)$$

The set of equations (3.16) gives the transition between column 2 and column 3 of Fig. 5.

C. Rules for Constructing Flow Charts

The continued expansion of Eq. (3.14) will result in a completed figure of the form of Fig. 5. The following general rules may be deduced for a generalized figure using 2^m time samples to obtain 2^m frequency samples:

1. A figure for 2^m samples will have $m+1$ columns of 2^m points each.
2. The points of the first column will be the original time samples arranged in order of the magnitude of their arguments: $k_{m-1} 2^{m-1} + k_{m-2} 2^{m-2} + \ldots + k_1 2 + k_0$.

3. The points of the second column will be points of an array $F(j_0,k_{m-2},\ldots,k_1,k_0)$. In general, the points of the r^{th} column will be points of an array $F(j_0,j_1,\ldots,j_{r-2},k_{m-r},k_{m-r-1},\ldots,k_1,k_0)$, arranged in order of the magnitude of the number $j_0 2^{m-1} + j_1 2^{m-2} + \ldots + j_{r-2} 2^{m-r+1} + k_{m-r} 2^{m-r} + k_{m-r-1} 2^{m-r-1} + \ldots + k_1 2 + k_0$.

4. All arrows leaving points in the r^{th} column for which the r^{th} index is zero will have weight 1 and will go horizontally to points in the $(r+1)^{st}$ column whose indices are the same, and to points whose indices are different in the r^{th} index only.

5. All arrows leaving points in the r^{th} column for which the r^{th} index is 1 will have weight $\exp\left[-i2\pi\left(j_{r-1}2^{-1} + j_{r-2}2^{-2} + \ldots + j_1 2^{-r+1} + j_0 2^{-r}\right)\right]$ and will go horizontally to points in the $(r+1)^{st}$ column whose indices are the same, and to points whose indices are different in the r^{th} index only. Note that the values of the weights are determined by j_0 through j_{r-1} which are the first r indices of the destinations of the arrows.

6. The points in the $(m+1)^{st}$ column will be the points of the spectrum $F_s(j_0,j_1,\ldots,j_{m-1})$ arranged in order of the magnitude of the number $j_0 2^{m-1} + j_1 2^{m-2} + \ldots + j_{m-2} 2^2 + j_{m-1}$. Note that the indices j_0 through j_{m-1} are reversed from their normal order.

D. Computational Savings

It can now be seen that by arranging the computations in the order described, considerable labor may be saved. In particular, rule 1 tells us that we have $m2^m$ points to calculate to obtain the spectrum as opposed to the $n^2 = (2^m)^2$ operations required to calculate the spectrum by direct application of Eq. (3.4). This computational algorithm was originally derived by Cooley and Tukey.

In Chapter VIII a new algorithm is derived which is even better suited to hardware implementation. Although the number of operations is not reduced, the manner in which they are performed in real time permits considerable savings in hardware.

E. Power Spectra

Up to this point we have been calculating the Fourier transform of the input waveform. In order to get the power spectrum of the input, we must take the square of the magnitude of the voltage transform already calculated. It is the power spectra rather than the voltage spectra that is the expected output from a "spectrum analyzer."

The power spectra may be arrived at in two ways. One is to form the autocorrelation function of the input and take the transform of the autocorrelation. The other way is to transform the input to get the voltage spectrum, and then take the magnitude squared of the voltage spectra to get the

power spectrum. We will choose the latter route as it will require far less in the way of additional circuitry.

The output of the real-time Fourier transform algorithm is a set of n complex numbers. Each of these numbers is of the form $A + iB$. The magnitude squared then will be given by $A^2 + B^2$. The squaring can be accomplished by putting A or B in both the x and y inputs of a multiplier such as the one designed in Chapter VI. Getting the magnitude squared will then require two multipliers and one adder for each of the n output samples.

Chapter IV

PASSBAND CHARACTERISTICS

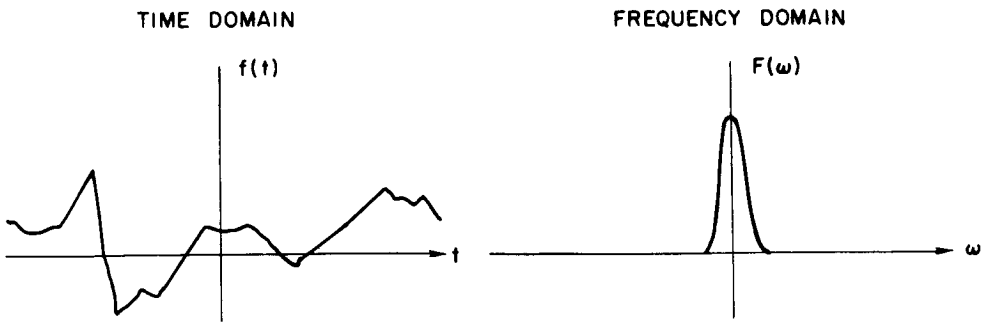
A. Infinite Number of Samples

The result of performing a discrete Fourier transform yields the transform of the sampled input signal rather than the transform of the original input.

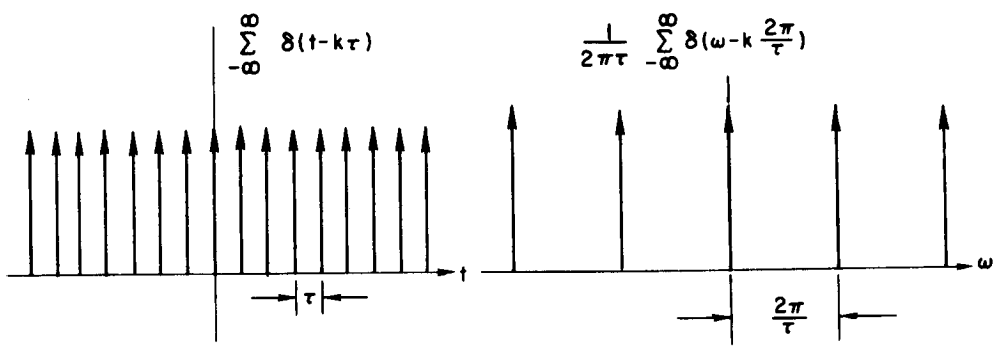
Figure 6 shows the derivation of the sampled signal's spectrum in terms of the spectrum of the unsampled signal. Here it is assumed that the sampling is done by an infinite string of impulses spaced τ seconds apart. The transform of this string of impulses is another string of impulses in the frequency domain of area $1/(2\pi\tau)$ spaced $2\pi/\tau$ radians apart. The sampling impulses and their transform are shown in Fig. 6b. Multiplying by the first string of impulses in the time domain is equivalent to convolution of the original frequency spectrum with the infinite string of impulses in the frequency domain as shown in Fig. 6c. Equations (4.1a) through (4.1c) describe the operations illustrated in Fig. 6. The convolution theorem is derived in Appendix A.

$$f(t) \longleftrightarrow F(\omega) \tag{4.1a}$$

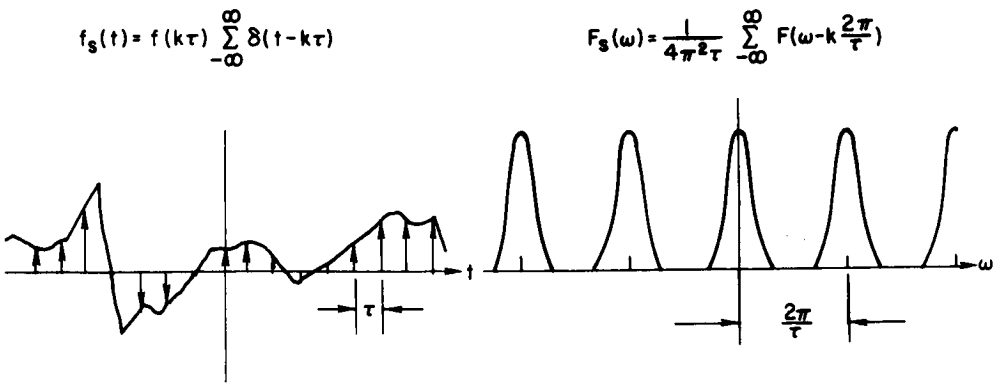
$$\sum_{-\infty}^{\infty} \delta(t - k\tau) \longleftrightarrow \frac{1}{2\pi\tau} \sum_{-\infty}^{\infty} \delta\left(\omega - k\,\frac{2\pi}{\tau}\right) \tag{4.1b}$$



a. f(t) and its transform F(ω)



b. Sampling impulses and their transform



c. Sampled $f_s(t)$ and its transform

Fig. 6. DERIVATION OF SPECTRUM OF SAMPLED f(t).

$$f_s(t) = f(t) \sum_{-\infty}^{\infty} \delta(t - k\tau) \longleftrightarrow F_s(\omega) \qquad (4.1c)$$

$$F_s(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega - \gamma) \frac{1}{2\pi\tau} \sum_{-\infty}^{\infty} \delta\left(\gamma - k\,\frac{2\pi}{\tau}\right) d\gamma$$

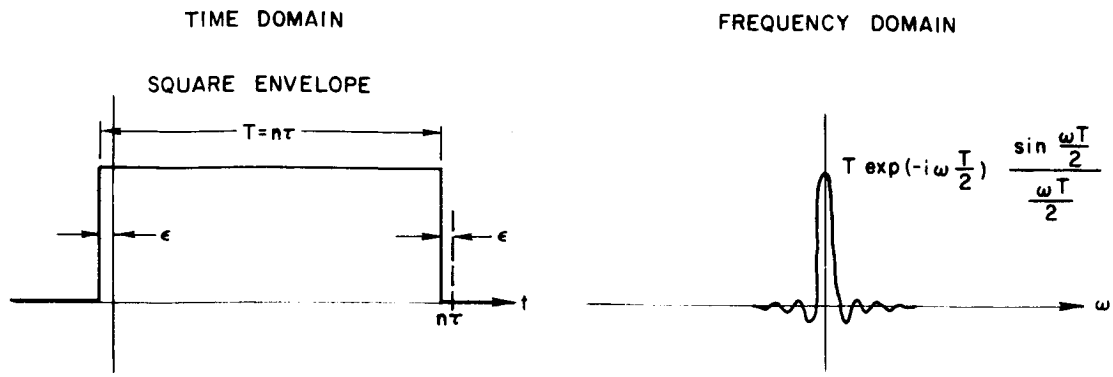
$$= \frac{1}{4\pi^2\tau} \sum_{-\infty}^{\infty} F\left(\omega - k\,\frac{2\pi}{\tau}\right)$$

As illustrated in Fig. 6c and Eq. (4.1c), $F_s(\omega)$, the spectrum of the sampled signal, is the sum of an infinite number of the original $F(\omega)$ with their centers spaced $2\pi/\tau$ apart along the frequency axis. If the spacing $2\pi/\tau$ is great enough, there will be no overlap of the individual spectra and the sampling has thus introduced no errors.

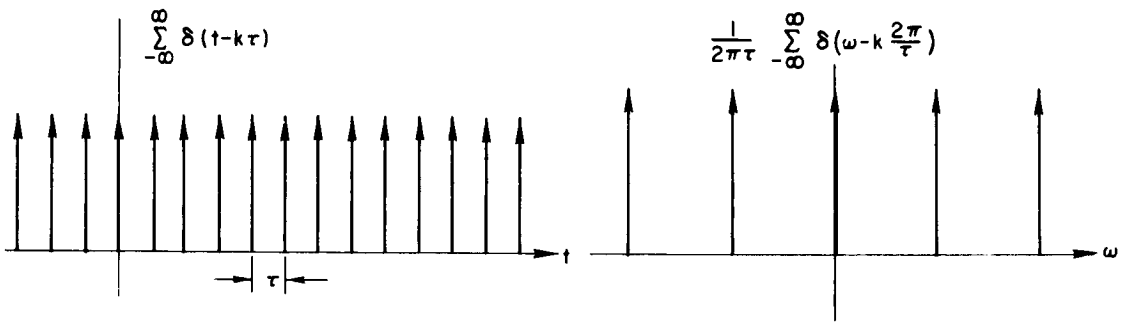
B. Finite Number of Samples

Up to now we have been talking about using an infinite number of samples in the time domain so that the convolution in the frequency domain is with an infinite string of impulses. Any finite machine we build must, of course, make calculations based on only a finite number of time samples. We now naturally ask what errors are introduced because we don't use an infinite number of samples.

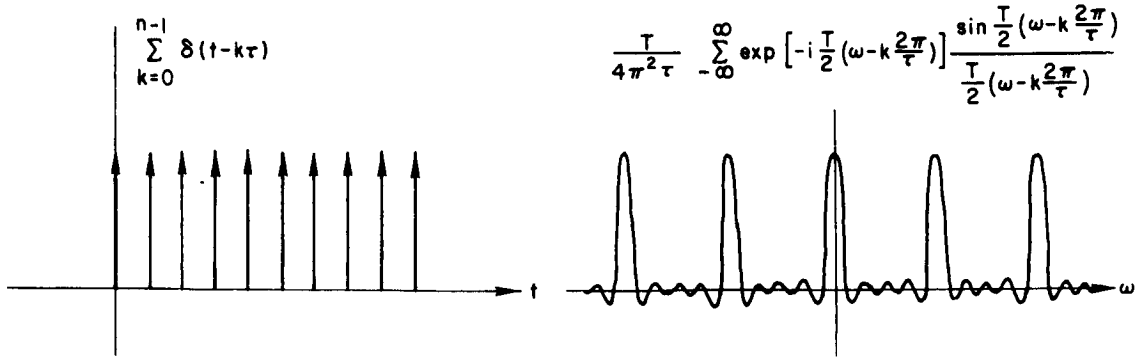
Figure 7 illustrates the derivation of our finite string of impulses by which we will multiply our time function $f(t)$ to get a finite number of samples $\hat{f}(t)$. $\hat{F}(\omega)$ will then be a convolution of $F(\omega)$ with the transform of the sampling function as shown in Fig. 7c.



a. Square envelope and its transform



b. Sampling impulses and their transform



c. Finite sampling impulse train and its transform

Fig. 7. DERIVATION OF SPECTRUM OF FINITE IMPULSE TRAIN.

To get a finite number of impulses we take the square window of Fig. 7a and multiply it by the infinite string of impulses in Fig. 7b. Note that the window is of width $n\tau$ and extends from $-\epsilon$ to $n\tau - \epsilon$ (ϵ being arbitrarily small) and thus will contain exactly n impulses.

The transform of the square window is

$$\exp\left[-i\omega\left(\frac{T}{2} - \epsilon\right)\right] T \frac{\sin (\omega T/2)}{\omega T/2} \tag{4.2a}$$

where for ϵ arbitrarily small, $\exp\left[-i\omega\left(\frac{T}{2} - \epsilon\right)\right]$ becomes $\exp\left(-i\omega \frac{T}{2}\right)$. When we multiply the window and the string of impulses in the time domain we must convolve

$$T \exp\left(-i\omega \frac{T}{2}\right) \frac{\sin (\omega T/2)}{\omega T/2} \tag{4.2b}$$

with a string of impulses in the frequency domain. The result is that the transform of our finite sampling function as shown in Fig. 7c is

$$\sum_{k=0}^{n-1} \delta(t - k\tau) \longleftrightarrow \frac{1}{4\pi^2\tau}$$

$$\sum_{k=-\infty}^{\infty} \exp\left[-i \frac{T}{2} \left(\omega - k \frac{2\pi}{\tau}\right)\right] T \frac{\sin \frac{T}{2} \left(\omega - k \frac{2\pi}{\tau}\right)}{\frac{T}{2} \left(\omega - k \frac{2\pi}{\tau}\right)} \tag{4.2c}$$

Since $\left|\exp\left[-i \frac{T}{2} \left(\omega - k \frac{2\pi}{\tau}\right)\right]\right| = 1$, a plot of the magnitude of the transform of our sampling function will look as shown in Fig. 7c. The spacing between the $(\sin x)/x$ functions is

$2\pi/\tau$. Remembering that $T = n\tau$, we find their height is $n/4\pi^2$ and their width varies inversely with n, the number of samples used.

When a time function $f(t)$ is sampled by multiplying by $\sum_{k=0}^{n-1} \delta(t - k\tau)$, then the spectrum of the sampled function $\hat{f}(t)$ can be expressed as the convolution of the original spectrum $F(\omega)$ with the spectrum of the sampling impulses as given by (4.2c). Thus

$$\hat{F}(\omega_0) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega_0 - \omega) \frac{n}{4\pi^2} \sum_{k=-\infty}^{\infty} \exp\left(-i \frac{n\tau}{2} \omega\right) \frac{\sin\left(\frac{n\tau}{2} \omega - nk\pi\right)}{\frac{n\tau}{2} \omega - nk\pi} \, d\omega \qquad (4.3)$$

It is actually $\hat{F}(\omega_0)$ which we are calculating by our discrete Fourier transform. From (4.3), $\hat{F}(\omega_0)$ is seen to be a weighted average of the true values of $F(\omega)$ in the vicinity of ω_0.

Let us assume that the sampling rate is high enough so that $F(\omega)$ is essentially zero for $-\pi/\tau \geq \omega \geq \pi/\tau$. This is the Nyquist rate. Then the limits on the integration in (4.3) may be changed as follows:

$$\int_{-\infty}^{\infty} \quad \text{becomes} \quad \int_{\omega=\omega_0-\pi/\tau}^{\omega=\omega_0+\pi/\tau} \qquad (4.4)$$

Since we are interested in $\hat{F}(\omega_0)$ only for $-\pi/\tau \leq \omega_0 \leq \pi/\tau$, the spectrum $F(\omega_0 - \omega)$ as shown in the convolution in Fig. 8 will overlap with only the $(\sin x)/x$ function centered at $\omega = 0$. Hence the summation in (4.3) will consist only of one term. Consequently we can write

$$\hat{F}(\omega_0) = \frac{n}{8\pi^3} \int_{-\infty}^{\infty} F(\omega_0 - \omega) \exp\left(-i \frac{n\tau}{2} \omega\right) \frac{\sin \frac{n\tau}{2} \omega}{\frac{n\tau}{2} \omega} d\omega \qquad (4.5)$$

In summary, it has been shown that the output of a digital spectrum analyzer computed using a finite number of samples

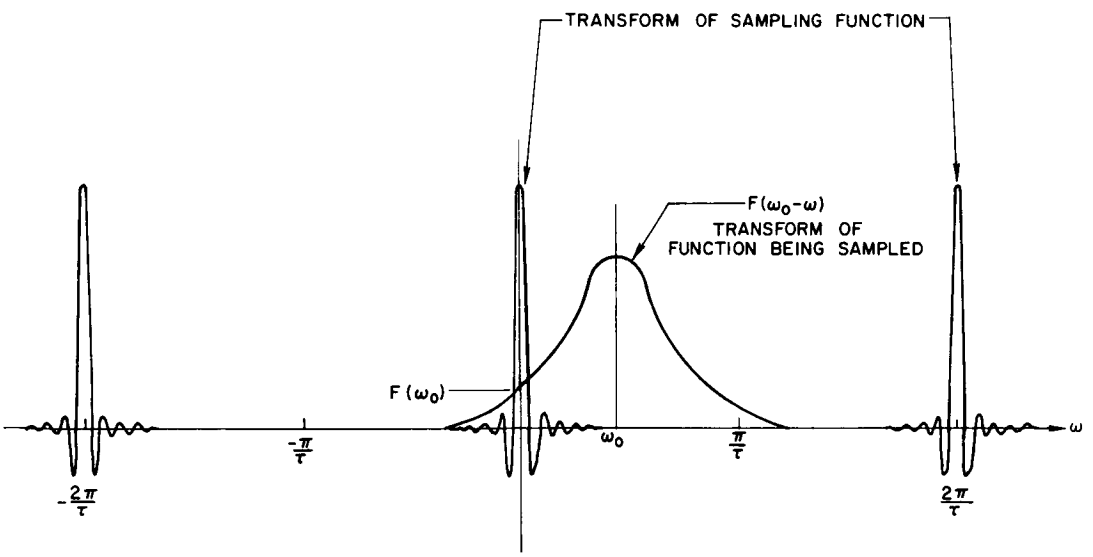


Fig. 8. CONVOLUTION OF $F(\omega)$ WITH SPECTRUM OF SAMPLING IMPULSE TRAIN.

will be an estimate $\hat{F}(\omega_0)$ of the true value of $F(\omega_0)$. The number $\hat{F}(\omega_0)$ obtained has been shown to be a weighted average of the true values of $F(\omega)$ in the vicinity of ω_0. The weighting function or "passband characteristic" is a $(\sin x)/x$ function whose width varies inversely with n, the number of samples used.

C. Multiplicative Constants

In Eq. (4.5), $\hat{F}(\omega_0)$ is a somewhat biased estimate of $F(\omega)$. Let's assume $F(\omega)$ is roughly constant and equal to $F(\omega_0)$ in the vicinity of ω_0; then Eq. (4.5) may be simplified to

$$\hat{F}(\omega_0) \approx \frac{nF(\omega_0)}{8\pi^3} \int_{-\infty}^{\infty} \exp\left(-i \frac{n\tau}{2} \omega\right) \frac{\sin \frac{n\tau}{2} \omega}{\frac{n\tau}{2} \omega} d\omega \quad (4.6)$$

Carrying through the integration we find

$$\hat{F}(\omega_0) \approx \frac{F(\omega_0)}{8\pi^2\tau} \quad (4.7)$$

Hence the calculated value of $\hat{F}(\omega_0)$ differs from the true value by a multiplicative constant of $1/(8\pi^2\tau)$. This multiplicative constant need not pose any problem depending on what use is made of the calculated spectrum. For instance, the digital spectrum may be reconverted to analog for an oscilloscope display of amplitude vs frequency. In this case the gain of the vertical amplifier of the oscilloscope may be adjusted to compensate for the multiplicative constant.

However, if a digital printout or other digital display is desired, the multiplicative constant will prove harder to deal with. The best solution in this case would probably be to use an amplifier with gain $8\pi^2\tau$ in front of the sampling process.

Chapter V

IMPROVING PASSBAND CHARACTERISTICS

A. Time-Domain Operations

In Eq. (4.5), the output of our digital spectrum analyzer for ω_0 was expressed as a weighted average of the true values of $F(\omega)$ in the vicinity of ω_0. The weighting function is to our digital spectrum analyzer what the passband characteristics are to an analog spectrum analyzer. As expressed in Eq. (4.5) the passband of our digital spectrum analyzer has the shape

$$\frac{n}{8\pi^3} \exp\left(-in \frac{\tau}{2} \omega\right) \frac{\sin \frac{n\tau}{2} \omega}{\frac{n\tau}{2} \omega} \qquad (5.1)$$

Note that this is precisely $1/(8\pi^3\tau)$ times the transform of the square envelope which we used to multiply the infinite sampling impulse train in order to obtain a finite number of sampling impulses.

This observation leads us to ask if we cannot modify the passband characteristics of our digital spectrum analyzer by changing the multiplying envelope. The answer is that indeed we can. In particular, if we use an envelope $h(t)$ which is $n\tau$ seconds wide, the output of our analyzer for ω_0 will be

$$\hat{F}(\omega_0) = \frac{1}{8\pi^3\tau} \int_{-\infty}^{\infty} F(\omega_0 - \omega)\, H(\omega)\, d\omega \qquad (5.2)$$

where $H(\omega)$ is the Fourier transform of $h(t)$.

We would like $H(\omega)$ to be infinitesimally narrow, i.e., an impulse. The only way this can be achieved is to make $h(t)$ infinitely wide. This, of course, is an impossibility for it means we would need an infinitely large machine to handle our infinite number of samples. The problem is analogous to the problem of shaping the far-field pattern of an antenna array by changing position and size of the array elements.

Our primary purpose in modifying the envelope $h(t)$ will be to improve our resolution in the frequency domain [i.e., to make $H(\omega)$ narrower]. Our secondary purpose will be to eliminate spurious responses which occur when two adjacent frequency components in the incoming signal are superimposed by the integral of (5.2) and either reinforce or cancel each other.

B. Easily Implemented Operations

Since we are doing our calculations digitally, certain $h(t)$ envelopes will be easier to achieve than others. In antennas, for instance, the elements of an array might be cosinusoidally weighted. Such a weighting would be difficult to realize digitally. Other arrays might be density weighted, which means that certain elements of the array are not driven or used. This would be easy to realize digitally since it merely means dropping some of the samples and replacing them by zero in the calculation. Also, any $h(t)$ in which the

samples are weighted by powers of 2 are easily realized, as this simply means shifting a binary number right or left by the appropriate number of bits. Any combination of weighting by zero and powers of 2 will be easy to realize digitally. It is the purpose of this section to explore these $h(t)$ and examine their effect on the resolution and response of our digital spectrum analyzer.

Let us then examine in more detail what sort of gains may be realized by using an envelope $h(t)$ which can be achieved simply by shifting the bits of the samples. If we digitize the inputs into p bits, then by shifting we can effect weights of 1, $1/2$, $1/4,\ldots,1/(2^{p-2})$, $1/(2^{p-1})$, and 0. Figure 9 illustrates a possible $h(t)$ achieved using these weights. It is easily seen that any envelopes of this type may be decomposed into a sum of a number of rectangular envelopes.

Since the Fourier transform is a linear operation, the transform of the sum is equal to the sum of the transforms of the individual rectangular envelopes. Consequently the composite $H(\omega)$ can be generated by adding together a number of $(\sin x)/x$ functions of different magnitude and different period.

In general, a rectangular envelope of height A, width B, and whose center is at C, will have a transform:

$$AB \exp(-i\omega C) \frac{\sin(\omega B/2)}{\omega B/2} \qquad (5.3)$$

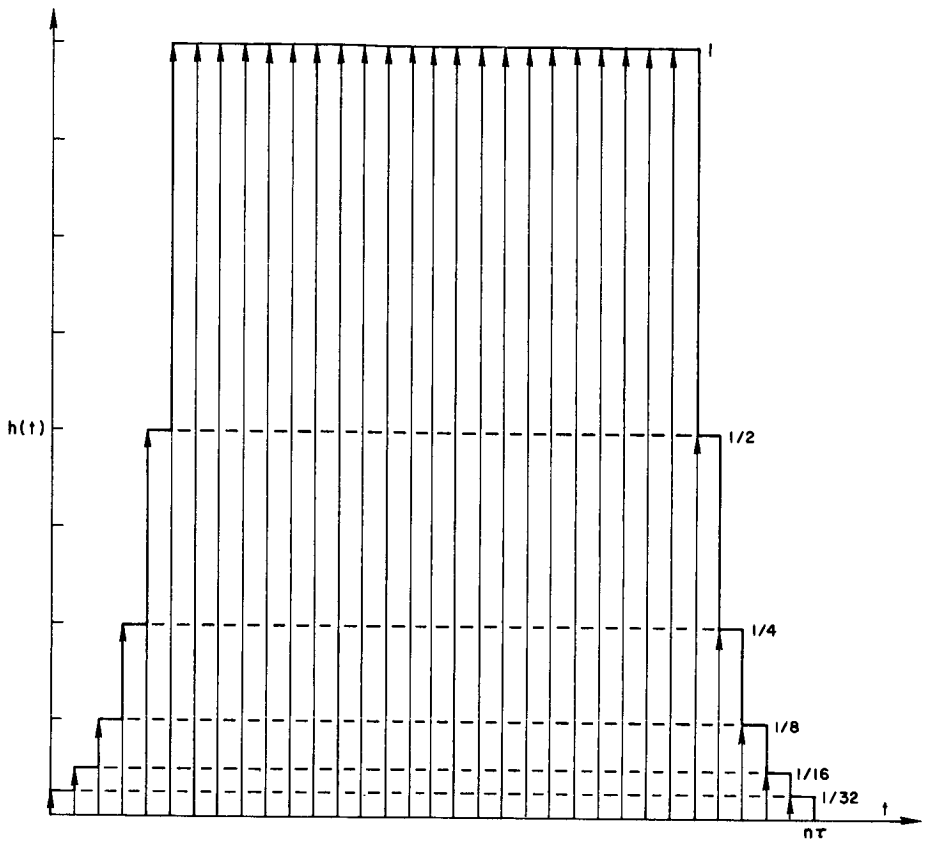


Fig. 9. TYPICAL h(t) ENVELOPE OBTAINED BY SHIFTING BITS OF DIGITIZED DATA INPUT.

A sum of such rectangular envelopes will have a transform:

$$\sum_j A_j B_j \exp(-i\omega C_j) \frac{\sin(\omega B_j/2)}{\omega B_j/2} \qquad (5.4)$$

Let us confine our attention to envelopes which are symmetrical about their centers, then the phase factor in (5.4) will be the same for all j and may be brought outside the summation.

$$\exp(-i\omega C) \sum_j A_j B_j \frac{\sin(\omega B_j/2)}{\omega B_j/2} \qquad (5.5)$$

The factor $A_j B_j$ is simply the area under the j^{th} rectangular envelope. We should note that $A_j = 1/2^j$ and $B_j \leq B_{j+1}$. Referring to Fig. 9 it is easy to see that each coefficient $A_k B_k$ is smaller than or equal to the sum of all the $A_j B_j$ for $j \geq k + 1$. This observation puts limits on the relative sizes of the various functions we are going to sum together.

Let us attempt to get a better feeling for the process that is occurring by working out the passband characteristics for the envelope shown in Fig. 9. Evaluating (5.5) we obtain

$$\exp(-i\omega\ 16\ \tau) \left(\tau \frac{\sin\ 16\ \omega\tau}{16\ \omega\tau} + \frac{15}{16} \tau \frac{\sin\ 15\ \omega\tau}{15\ \omega\tau} \right.$$

$$+ \frac{7}{4} \tau \frac{\sin\ 14\ \omega\tau}{14\ \omega\tau} + \frac{13}{4} \tau \frac{\sin\ 13\ \omega\tau}{13\ \omega\tau}$$

$$\left. + 6\tau \frac{\sin\ 12\ \omega\tau}{12\ \omega\tau} + 11\tau \frac{\sin\ 11\ \omega\tau}{11\ \omega\tau} \right) \qquad (5.6)$$

The passband characteristics expressed in Eq. (5.6) are plotted in Fig. 10 and can be compared to the passband characteristics of the original rectangular $h(t)$ which are plotted in the same figure. The result is a passband which is generally wider and lower than the original. Similar

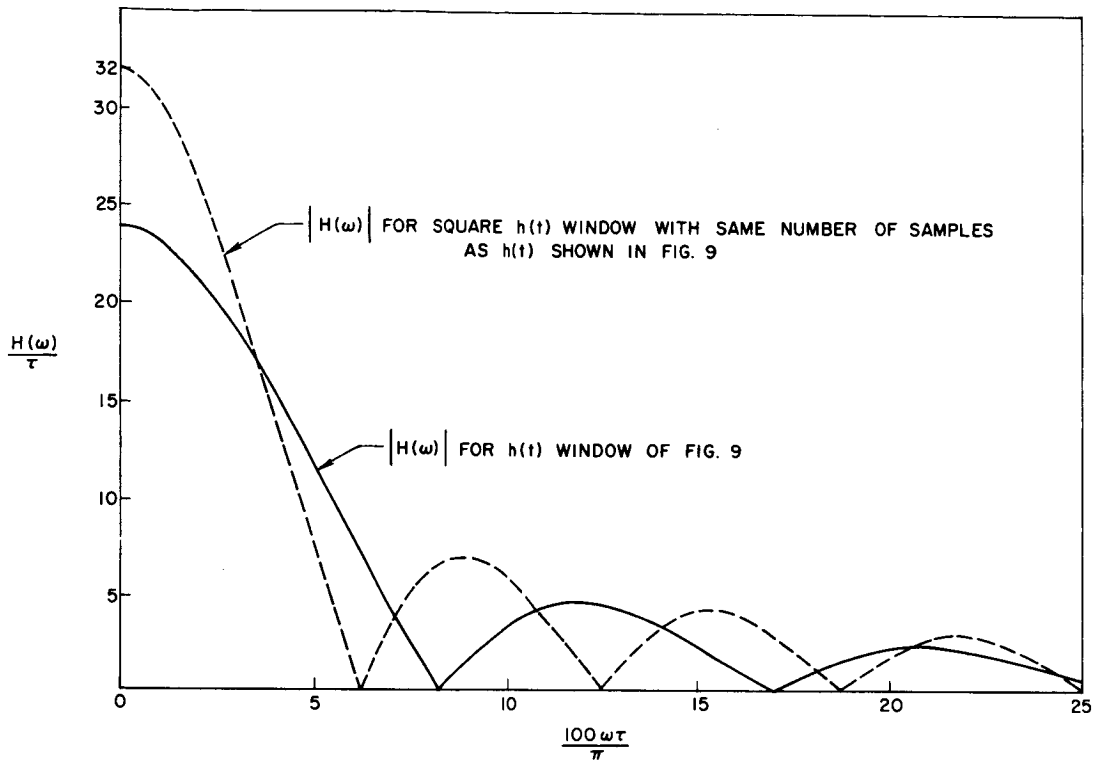


Fig. 10. MAGNITUDE OF $H(\omega)$ FOR AN EASILY MECHANIZED $h(t)$.

results are obtained for other simply implemented $h(t)$ which require only a shifting of the bits of the input samples.

C. Frequency-Domain Operations

Some operations which prove difficult to implement in the time domain may be much more simply performed in the frequency domain. Suppose, for instance, that we simply add to each frequency estimate the value of the two adjacent estimates multiplied by a constant K. The estimate at ω_0 can be written as

$$\hat{F}(\omega_0) = \frac{n}{8\pi^3} \int_{-\infty}^{\infty} F(\omega) \exp\left[-i \frac{n\tau}{2} (\omega_0 - \omega)\right] \frac{\sin \frac{n\tau}{2} (\omega_0 - \omega)}{\frac{n\tau}{2} (\omega_0 - \omega)} d\omega \tag{5.7}$$

The adjacent estimates are centered at $\omega_0 \pm (2\pi/n\tau)$ and can be written as

$$K\hat{F}\left(\omega_0 \pm \frac{2\pi}{n\tau}\right) = \frac{Kn}{8\pi^3} \int_{-\infty}^{\infty} F(\omega) \exp\left[-i \frac{n\tau}{2} \left(\omega_0 \pm \frac{2\pi}{n\tau} - \omega\right)\right]$$

$$\cdot \frac{\sin \frac{n\tau}{2} \left(\omega_0 \pm \frac{2\pi}{n\tau} - \omega\right)}{\frac{n\tau}{2} \left(\omega_0 \pm \frac{2\pi}{n\tau} - \omega\right)} d\omega \tag{5.8}$$

Using the identity

$$e^{-i\pi} = -1 \tag{5.9}$$

we may express (5.8) as

$$K\hat{F}\left(\omega_0 \pm \frac{2\pi}{n\tau}\right) = -\frac{Kn}{8\pi^3} \int_{-\infty}^{\infty} F(\omega) \exp\left[-i \frac{n\tau}{2} (\omega_0 - \omega)\right]$$

$$\cdot \frac{\sin \frac{n\tau}{2} \left(\omega_0 \pm \frac{2\pi}{n\tau} - \omega\right)}{\frac{n\tau}{2} \left(\omega_0 \pm \frac{2\pi}{n\tau} - \omega\right)} d\omega \tag{5.10}$$

Combining (5.7) and (5.10) yields

$$K\hat{F}\left(\omega_0 - \frac{2\pi}{n\tau}\right) + \hat{F}(\omega_0) + K\hat{F}\left(\omega_0 + \frac{2\pi}{n\tau}\right)$$

$$= \hat{\hat{F}}(\omega_0) = \frac{n}{8\pi^3} \int_{-\infty}^{\infty} F(\omega) \exp\left[-i \frac{n\tau}{2} (\omega_0 - \omega)\right]$$

$$\cdot \left[-K \frac{\sin \frac{n\tau}{2} \left(\omega_0 - \frac{2\pi}{n\tau} - \omega\right)}{\frac{n\tau}{2} \left(\omega_0 - \frac{2\pi}{n\tau} - \omega\right)} + \frac{\sin \frac{n\tau}{2} (\omega_0 - \omega)}{\frac{n\tau}{2} (\omega_0 - \omega)}\right.$$

$$\left. - K \frac{\sin \frac{n\tau}{2} \left(\omega_0 + \frac{2\pi}{n\tau} - \omega\right)}{\frac{n\tau}{2} \left(\omega_0 + \frac{2\pi}{n\tau} - \omega\right)}\right] d\omega \qquad (5.11)$$

The spectral estimate $\hat{\hat{F}}(\omega_0)$ as given in (5.11) is seen to be equal to the true spectra $F(\omega)$ of the input weighted by a passband characteristic which is the sum of three shifted $(\sin x)/x$ type functions. Figure 11 illustrates the three $(\sin x)/x$ functions. For ease of illustration the complex exponential (whose magnitude is one) has not been shown. The value of K can be adjusted so that the two adjacent $(\sin x)/x$ functions will partially cancel the sidelobes on the central $(\sin x)/x$ function. Perini [1964] studied a mathematically similar problem in the shaping of the far-field patterns of linear antenna arrays.

Figures 12 through 14 show the combined passband characteristic for several interesting values of K. Figure 12 is for $K = 0$, which gives the original $(\sin x)/x$ passband. Figure 13 is for $K = -0.426$, which yields a zero at the peak of the first sidelobe of the original passband. This choice

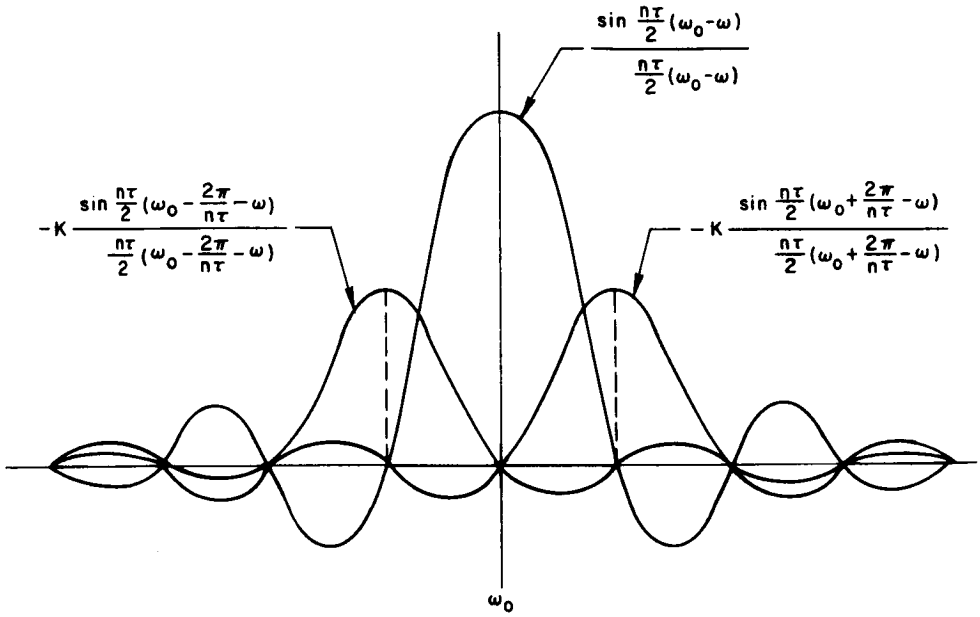


Fig. 11. ADDITION OF THREE ADJACENT WEIGHTED SPECTRAL ESTIMATES.

of K makes the close-in sidelobes very small.

Figure 14 shows the passband for K = -0.5. This particular choice of K gives sidelobes which fall off as $1/(\omega_0 - \omega)^3$ rather than $1/\omega_0 - \omega$. To show this we must expand Eq. (5.11) using $\sin(x \pm \pi) = -\sin x$:

$$\hat{\hat{F}}(\omega_0) = \frac{n}{8\pi^3} \int_{-\infty}^{\infty} F(\omega) \exp\left[-i \frac{n\tau}{2} (\omega_0 - \omega)\right] \frac{\sin \frac{n\tau}{2} (\omega_0 - \omega)}{\frac{n\tau}{2}}$$

$$\cdot \left(\frac{K}{\omega_0 - \frac{2\pi}{n\tau} - \omega} + \frac{1}{\omega_0 - \omega} + \frac{K}{\omega_0 + \frac{2\pi}{n\tau} - \omega} \right) d\omega \qquad (5.12)$$

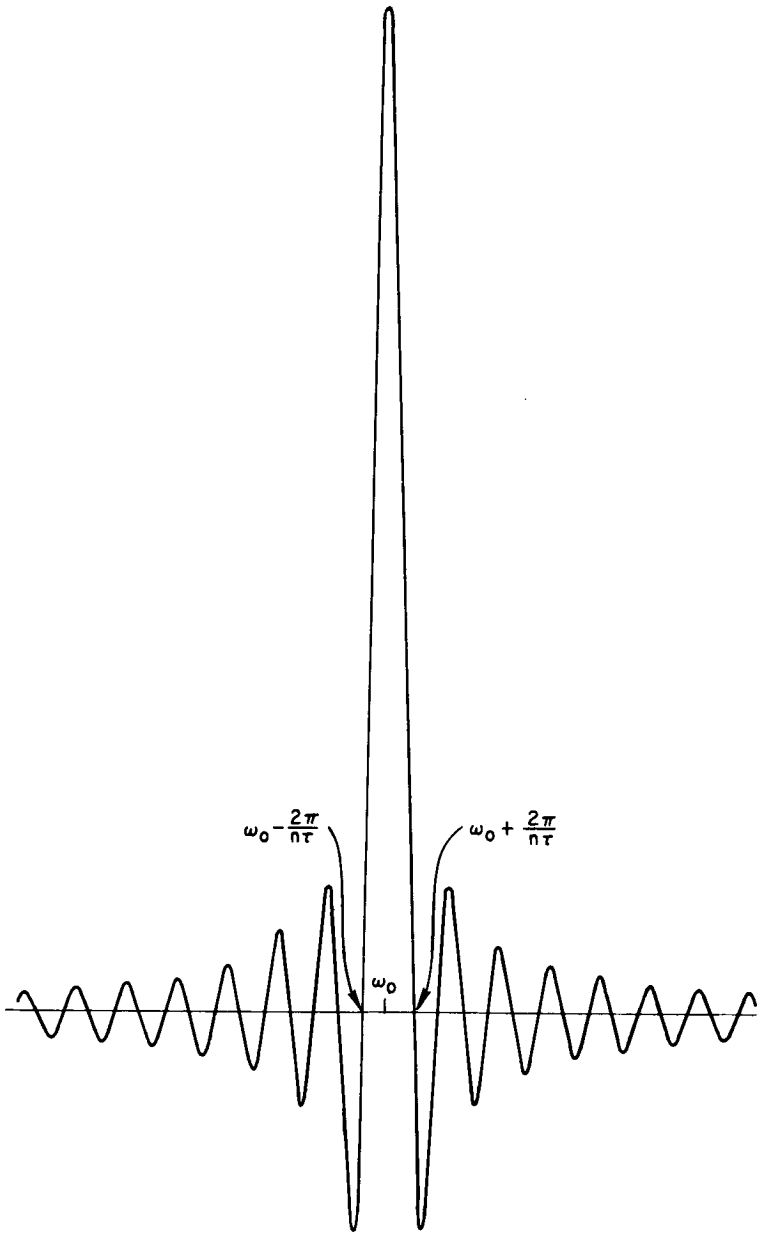


Fig. 12. PASSBAND FOR K = 0.

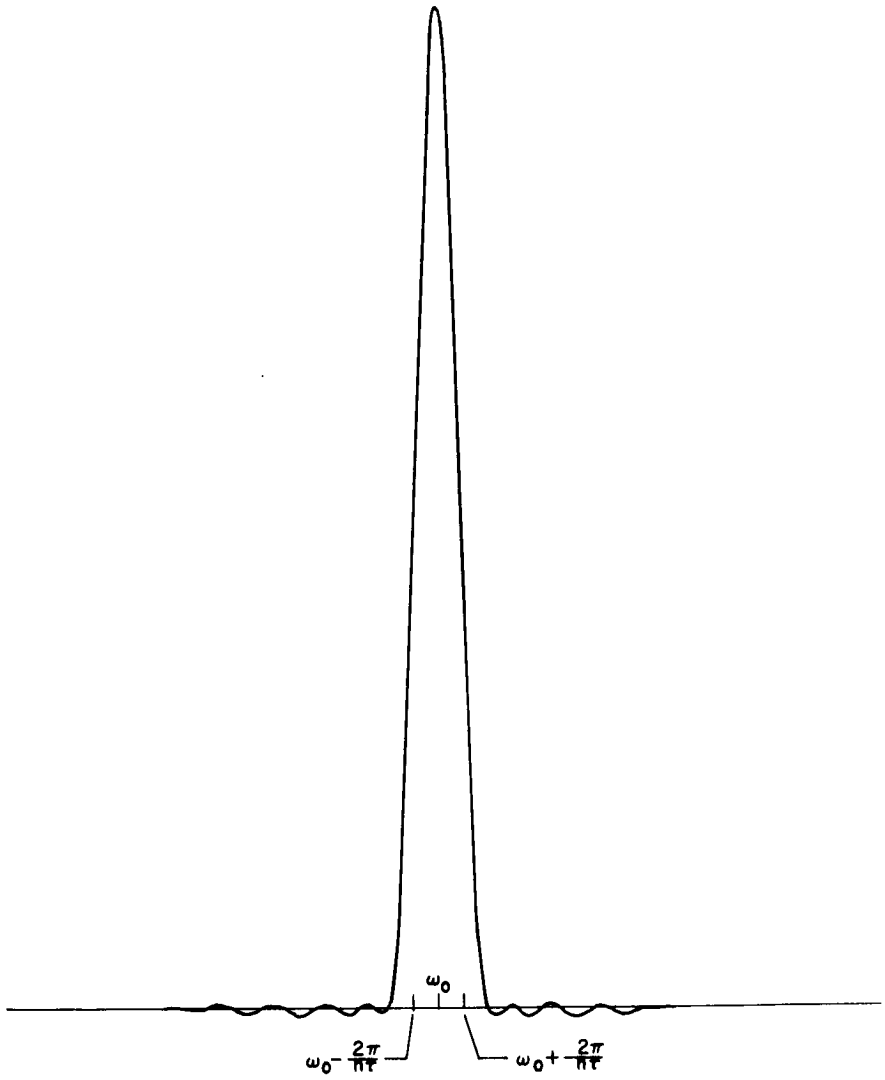


Fig. 13. PASSBAND FOR K = -0.426.

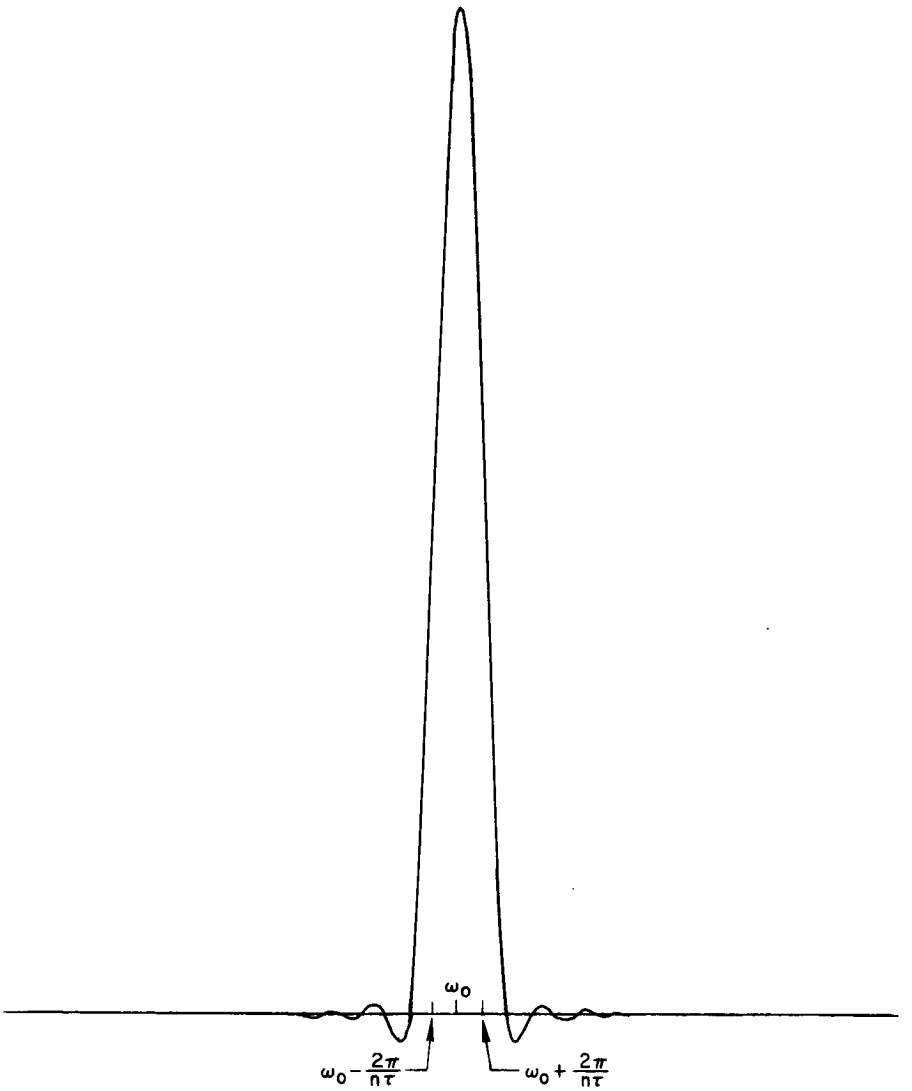


Fig. 14. PASSBAND FOR K = -0.5.

Putting (5.12) over a common denominator gives

$$\hat{\hat{F}}(\omega_0) = \frac{n}{8\pi^3} \int_{-\infty}^{\infty} F(\omega) \exp\left[-i \frac{n\tau}{2} (\omega_0 - \omega)\right] \frac{\sin \frac{n\tau}{2} (\omega_0 - \omega)}{\frac{n\tau}{2}}$$

$$\cdot \left[\frac{(2K + 1)(\omega_0 - \omega)^2 - \left(\frac{2\pi}{n\tau}\right)^2}{(\omega_0 - \omega)^3 - \left(\frac{2\pi}{n\tau}\right)^2 (\omega_0 - \omega)}\right] d\omega \qquad (5.13)$$

If we now set $K = -1/2$ we obtain

$$\hat{\hat{F}}(\omega_0) = \frac{n}{8\pi^3} \int_{-\infty}^{\infty} F(\omega) \exp\left[-i \frac{n\tau}{2} (\omega_0 - \omega)\right] \sin \frac{n\tau}{2} (\omega_0 - \omega)$$

$$\cdot \left[- \frac{\pi^2}{\left(\frac{n\tau}{2}\right)^3 (\omega_0 - \omega)^3 - \pi^2 \frac{n\tau}{2} (\omega_0 - \omega)}\right] d\omega \qquad (5.14)$$

For ω near ω_0, the $(\omega_0 - \omega)^3$ term is small and (5.14) can be written

$$\hat{\hat{F}}(\omega_0) \approx \frac{n}{8\pi^3} \int_{-\infty}^{\infty} F(\omega) \exp\left[-i \frac{n\tau}{2} (\omega_0 - \omega)\right] \frac{\sin \frac{n\tau}{2} (\omega_0 - \omega)}{\frac{n\tau}{2} (\omega_0 - \omega)} d\omega$$

$$\text{for} \quad \left|\left(\frac{n\tau}{2}\right)^2 (\omega_0 - \omega)^3\right| << \left|\pi^2 (\omega_0 - \omega)\right|$$

$$\text{and} \quad K = -\frac{1}{2} \qquad (5.15)$$

For ω far from ω_0, the $\omega_0 - \omega$ term is negligible and (5.14) can be written

$$\hat{\hat{F}}(\omega_0) = -\frac{n}{8\pi}\int_{-\infty}^{\infty} F(\omega)\exp\left[-i\frac{n\tau}{2}(\omega_0-\omega)\right]\frac{\sin\frac{n\tau}{2}(\omega_0-\omega)}{\left[\frac{n\tau}{2}(\omega_0-\omega)\right]^3}\,d\omega$$

$$\text{for}\quad \left|\left(\frac{n\tau}{2}\right)^2(\omega_0-\omega)^3\right| >> \left|\pi^2(\omega_0-\omega)\right|$$

$$\text{and}\quad K = -\frac{1}{2} \qquad (5.16)$$

Equation (5.16) explicitly points up the $1/(\omega_0 - \omega)^3$ aprons of the passband characteristic for $K = -1/2$. Q.E.D.

The addition of $\hat{F}(\omega_0)$ to K times the two adjacent frequency estimates is an easily implemented modification and has been shown above to result in substantially reduced sidelobes in the passband. The special case of $K = -1/2$ is especially easy to implement as no multiplication is needed. The factor of 1/2 is achieved by hardwiring a shift of 1 bit. This special case has been dubbed "hanning" by Blackman and Tukey [1959].

D. Equivalent Time-Domain Operation

The summing in the frequency domain could be replaced by an equivalent time-domain operation. As it turns out, such an operation is far more complicated to implement from a hardware standpoint. To find out what the equivalent time-domain operation is, we first note that the summing of the three $(\sin x)/x$ functions can be expressed as a convolution of a single $(\sin x)/x$ function with three impulses. Expanding the terms in brackets in (5.11) yields

$$- K \frac{\sin \frac{n\tau}{2}\left(\omega_0 - \frac{2\pi}{n\tau} - \omega\right)}{\frac{n\tau}{2}\left(\omega_0 - \frac{2\pi}{n\tau} - \omega\right)} + \frac{\sin \frac{n\tau}{2}(\omega_0 - \omega)}{\frac{n\tau}{2}(\omega_0 - \omega)} - K \frac{\sin \frac{n\tau}{2}\left(\omega_0 + \frac{2\pi}{n\tau} - \omega\right)}{\frac{n\tau}{2}\left(\omega_0 + \frac{2\pi}{n\tau} - \omega\right)}$$

$$= \int_{-\infty}^{\infty} \left[-K\delta\left(\Omega - \frac{2\pi}{n\tau}\right) + \delta(\Omega) - K\delta\left(\Omega + \frac{2\pi}{n\tau}\right)\right]\left[\frac{\sin \frac{n\tau}{2}(\omega_0 - \omega - \Omega)}{\frac{n\tau}{2}(\omega_0 - \omega - \Omega)}\right] d\Omega \tag{5.17}$$

Plugging (5.17) back into (5.11) gives

$$\hat{\hat{F}}(\omega_0) = \frac{n}{8\pi^3} \int_{-\infty}^{\infty} F(\omega) \exp\left[-i \frac{n\tau}{2}(\omega_0 - \omega)\right] \int_{-\infty}^{\infty} \left[-K\delta\left(\Omega - \frac{2\pi}{n\tau}\right)\right.$$

$$\left. + \delta(\Omega) - K\delta\left(\Omega + \frac{2\pi}{n\tau}\right)\right]\left[\frac{\sin \frac{n\tau}{2}(\omega_0 - \omega - \Omega)}{\frac{n\tau}{2}(\omega_0 - \omega - \Omega)}\right] d\Omega \, d\omega \tag{5.18}$$

By reversing the order of integration and pulling out a factor of $\exp\left(-i \frac{n\tau}{2} \Omega\right)$ we obtain

$$\hat{\hat{F}}(\omega_0) = \frac{n}{8\pi^3} \int_{-\infty}^{\infty} \left[-K\delta\left(\Omega - \frac{2\pi}{n\tau}\right) + \delta(\Omega) - K\delta\left(\Omega + \frac{2\pi}{n\tau}\right)\right] \exp\left(-i \frac{n\tau}{2} \Omega\right)$$

$$\cdot \int_{-\infty}^{\infty} F(\omega) \exp\left[-i \frac{n\tau}{2}(\omega_0 - \omega - \Omega)\right]\left[\frac{\sin \frac{n\tau}{2}(\omega_0 - \omega - \Omega)}{\frac{n\tau}{2}(\omega_0 - \omega - \Omega)}\right] d\omega \, d\Omega \tag{5.19}$$

Referring to (5.7), the inner integral of (5.19) is seen to be $(8\pi^3)/n \ \hat{F}(\omega_0 - \Omega)$.

Consequently we find

$$\hat{\hat{F}}(\omega_0) = \int_{-\infty}^{\infty} \left[-K\delta\left(\Omega - \frac{2\pi}{n\tau}\right) + \delta(\Omega) - K\delta\left(\Omega + \frac{2\pi}{n\tau}\right)\right]$$

$$\exp\left(-i\,\frac{n\tau}{2}\,\Omega\right)\,\hat{F}(\omega_0 - \Omega)\,d\Omega \qquad (5.20)$$

Equation (5.20) is in the form of a convolution in the frequency domain which is equivalent to a multiplication in the time domain by an $h(t)$ window function whose transform is given by

$$h(t) \longleftrightarrow 2\pi\left\{\delta(\omega) - K\left[\delta\left(\omega - \frac{2\pi}{n\tau}\right) + \delta\left(\omega + \frac{2\pi}{n\tau}\right)\right]\right\} \exp\left(-i\,\frac{n\tau}{2}\,\omega\right) \qquad (5.21)$$

The window $h(t)$ then can be determined to be

$$h(t) = 1 - 2K\cos\frac{2\pi}{n\tau}\left(t - \frac{n\tau}{2}\right)$$

$$= 1 + 2K\cos\frac{2\pi t}{n\tau} \qquad (5.22)$$

Figure 15 shows the effect of this window function on the input data samples for a K value of -0.426.

Realization of this increase in resolution by multiplying the n input samples by the appropriate $h(t)$ will require n complex multiplications. Realization of the increase by working in the frequency domain will require n complex multipliers and $2n$ complex adders as shown in Fig. 16. However, if K is set at -1/2, then the multiplications can be replaced by hardwired shifting and complementing and we are left simply with $2n$ adders. Consequently, working in the

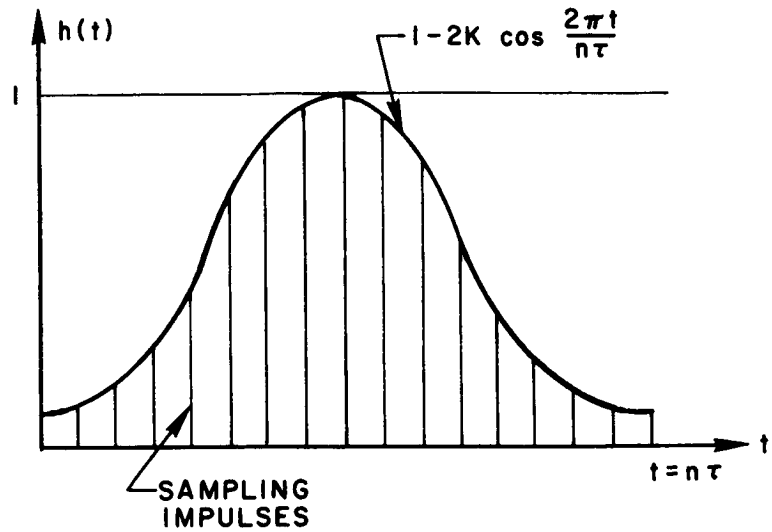
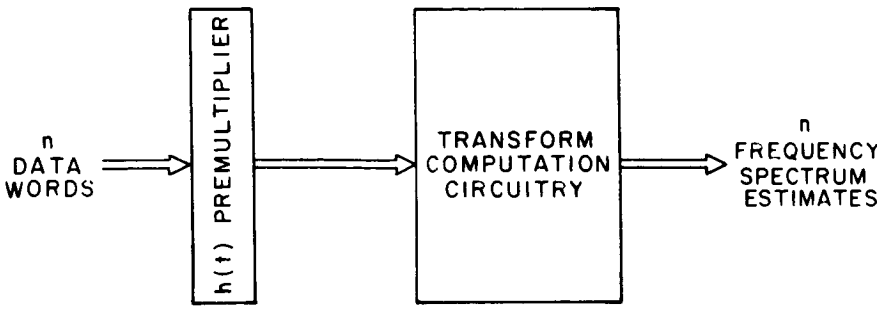


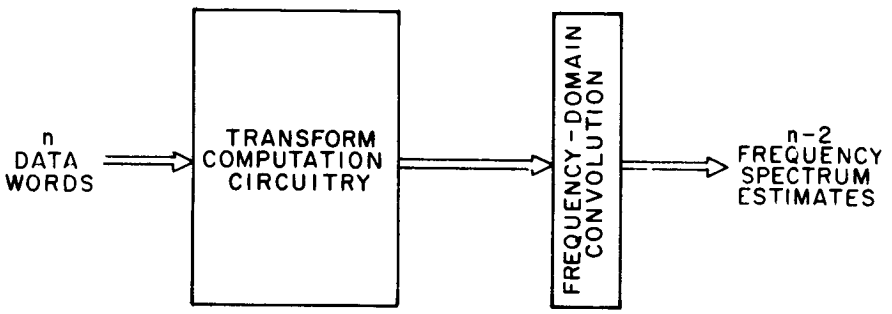
Fig. 15. TIME-DOMAIN WINDOW EQUIVALENT TO WEIGHTED SUMMATION OF ADJACENT FREQUENCY-SPECTRUM ESTIMATES.

frequency domain here is really only attractive if the value of K can be set to -1/2 or some other easily realized constant multiplier.

A new real-time, hardware-oriented algorithm is derived in Chapter VIII. In using this algorithm the input is never in a stationary array, and hence premultiplication by an appropriate h(t) window is further complicated. It is for this reason and for the reasons in the above paragraph that we will choose to do our modifications in the frequency domain at the output of the computation circuitry. We will also choose a value of K = -1/2 for ease of implementation as mentioned above. The resulting windowing circuitry is shown in Fig. 17.



a. Premultiplication by h(t) window



b. Equivalent frequency-domain convolution

Fig. 16. EQUIVALENT TIME- AND FREQUENCY-DOMAIN OPERATION.

E. Double Hanning

A second hanning has been used by some researchers in order to improve the resolution further. Let us call the double-hanned signal $\hat{\hat{\hat{F}}}(\omega_0)$. We can then write

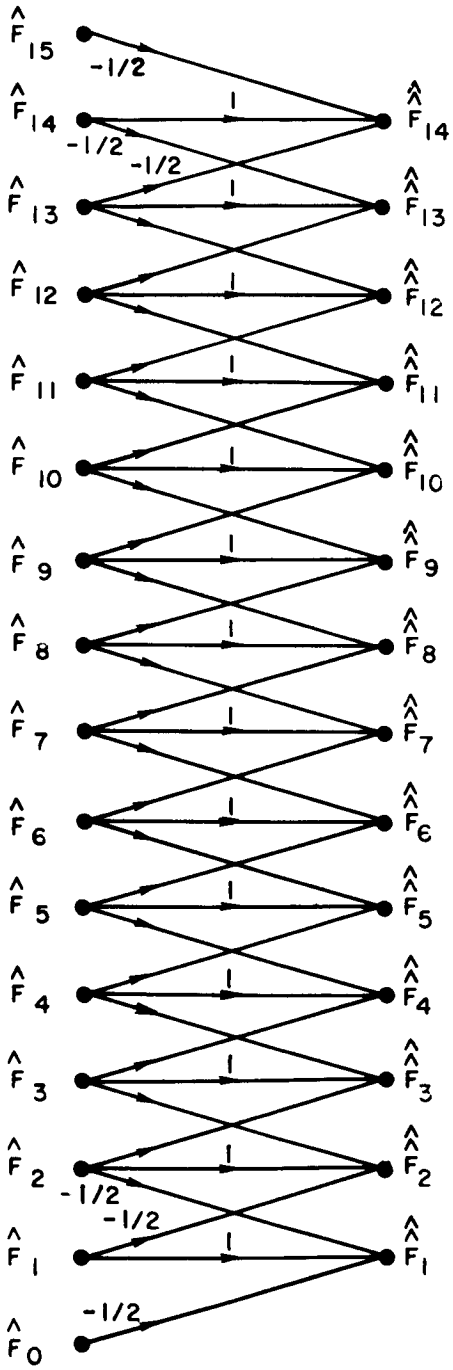


Fig. 17. FLOW GRAPH FOR HANNING OPERATION.

$$\hat{\hat{\hat{F}}}(\omega_0) = -\frac{1}{2}\hat{\hat{F}}\left(\omega_0 - \frac{2\pi}{n\tau}\right) + \hat{\hat{F}}(\omega_0) - \frac{1}{2}\hat{\hat{F}}\left(\omega_0 + \frac{2\pi}{n\tau}\right) \qquad (5.23)$$

Using Eq. (5.11) for $\hat{\hat{F}}(\omega_0)$ yields

$$\begin{aligned}\hat{\hat{\hat{F}}}(\omega_0) = &-\frac{1}{2}\left\{-\frac{1}{2}\hat{F}\left(\omega_0 - \frac{4\pi}{n\tau}\right) + \hat{F}\left(\omega_0 - \frac{2\pi}{n\tau}\right) - \frac{1}{2}\hat{F}(\omega_0)\right\} \\ &+ \left\{-\frac{1}{2}\hat{F}\left(\omega_0 - \frac{2\pi}{n\tau}\right) + \hat{F}(\omega_0) - \frac{1}{2}\hat{F}\left(\omega_0 + \frac{2\pi}{n\tau}\right)\right\} \\ &- \frac{1}{2}\left\{-\frac{1}{2}\hat{F}(\omega_0) + \hat{F}\left(\omega_0 + \frac{2\pi}{n\tau}\right) - \frac{1}{2}\hat{F}\left(\omega_0 + \frac{4\pi}{n\tau}\right)\right\} \qquad (5.24)\end{aligned}$$

Collecting like terms

$$\begin{aligned}\hat{\hat{\hat{F}}}(\omega_0) = &\frac{1}{4}\hat{F}\left(\omega_0 - \frac{4\pi}{n\tau}\right) - \hat{F}\left(\omega_0 - \frac{2\pi}{n\tau}\right) + \frac{3}{2}\hat{F}(\omega_0) \\ &- \hat{F}\left(\omega_0 + \frac{2\pi}{n\tau}\right) + \frac{1}{4}\hat{F}\left(\omega_0 + \frac{4\pi}{n\tau}\right) \qquad (5.25)\end{aligned}$$

We may further express $\hat{F}(\omega_0)$ in terms of the true $F(\omega)$, yielding

$$\begin{aligned}\hat{\hat{\hat{F}}}(\omega_0) = \frac{n}{8\pi^3}\int_{-\infty}^{\infty} F(\omega)\exp\left[-i\frac{n\tau}{2}(\omega_0 - \omega)\right]\Bigg[&\frac{1}{4}\frac{\sin\frac{n\tau}{2}\left(\omega_0 - \omega - \frac{4\pi}{n\tau}\right)}{\frac{n\tau}{2}\left(\omega_0 - \omega - \frac{4\pi}{n\tau}\right)} \\ &+ \frac{\sin\frac{n\tau}{2}\left(\omega_0 - \omega - \frac{2\pi}{n\tau}\right)}{\frac{n\tau}{2}\left(\omega_0 - \omega - \frac{2\pi}{n\tau}\right)} + \frac{3}{2}\frac{\sin\frac{n\tau}{2}(\omega_0 - \omega)}{\frac{n\tau}{2}(\omega_0 - \omega)} \\ &+ \frac{\sin\frac{n\tau}{2}\left(\omega_0 - \omega + \frac{2\pi}{n\tau}\right)}{\frac{n\tau}{2}\left(\omega_0 - \omega + \frac{2\pi}{n\tau}\right)} + \frac{1}{4}\frac{\sin\frac{n\tau}{2}\left(\omega_0 - \omega + \frac{4\pi}{n\tau}\right)}{\frac{n\tau}{2}\left(\omega_0 - \omega + \frac{4\pi}{n\tau}\right)}\Bigg] d\omega \qquad (5.26)\end{aligned}$$

From (5.26) we see that the double-hanned spectral estimate $\hat{\hat{\hat{F}}}(\omega_0)$ has a passband characteristic which is the sum of five shifted (sin x)/x-type functions. Since it was not immediately obvious that this passband was going to be an improvement over the single-hanned case, a computer program was written to compute and plot this passband. The results of this plot can be seen in Fig. 18.

Although the sidelobes do drop off more rapidly than the single-hanned passband, the central peak has been made considerably broader. The improvement in passband characteristics in going from single hanning (Fig. 14) to double hanning (Fig. 18) is much less marked than is the improvement in going from no hanning (Fig. 12) to single hanning. Since the windowing circuitry must be approximately doubled to implement double hanning, it is felt that the single-hanned frequency estimate is the best compromise.

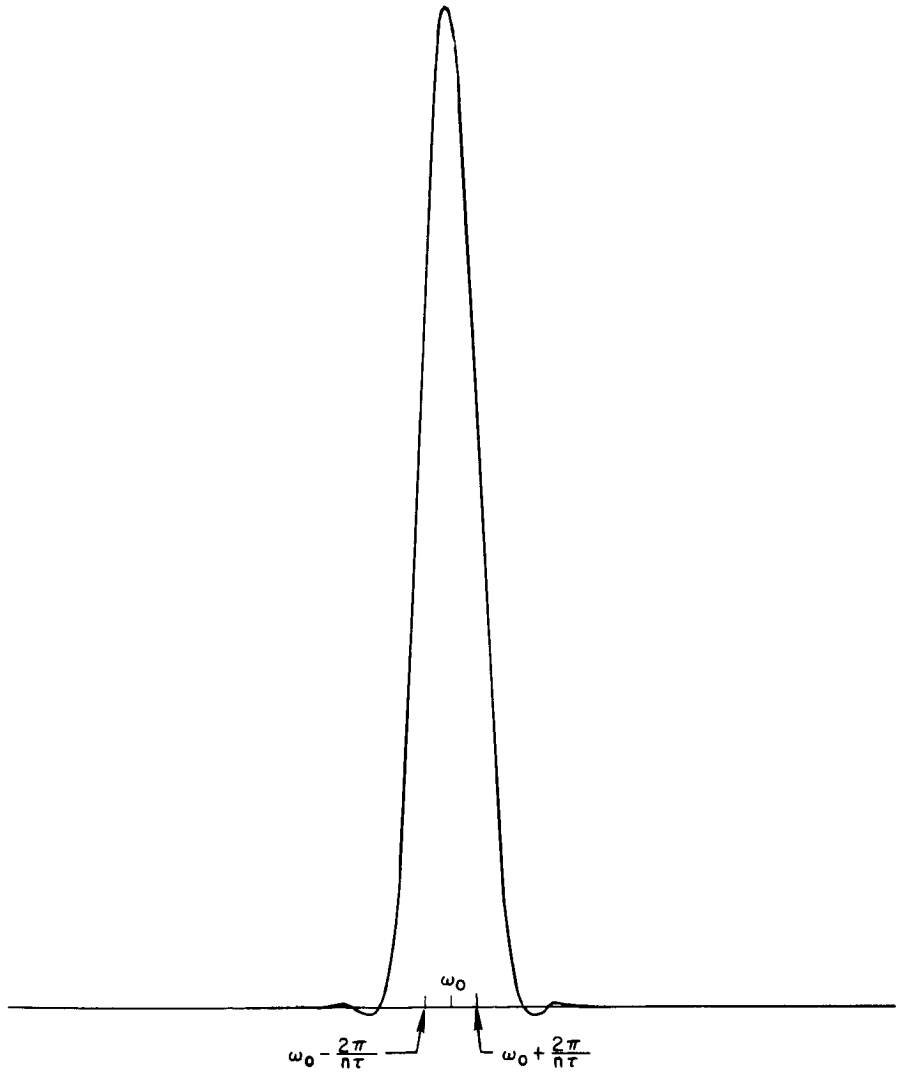


Fig. 18. PASSBAND ACHIEVED BY HANNING TWICE.

Chapter VI

HARDWARE IMPLEMENTATION OF FOURIER TRANSFORM ALGORITHM

A. Basic Functional Building Block

Let us now investigate the problems of mechanizing the Cooley-Tukey algorithm. For $n = 2^m$ input samples the flow chart of the calculations will contain $m+1$ columns representing m stages of calculation plus the original sample points. Each of the points in the $(r+1)^{st}$ column will be the complex sum of one of the numbers in the r^{th} column and another of the numbers in the r^{th} column multiplied by a complex constant.

Generally speaking, all the numbers dealt with will have a real part and an imaginary part. Hence all additions will, in fact, be two additions (Real + Real; Imaginary + Imaginary) and all multiplications will be four multiplications (Re × Re; Re × Im; Im × Re; Im × Im).

Let us group together in the r^{th} column pairs of points which differ only in the r^{th} index. If we do this, we note that each such pair of points affects only the corresponding pair of points in the $(r+1)^{st}$ column. Furthermore, the pair of points in the $(r+1)^{st}$ column depends on no other points in the r^{th} column but this pair. This is a very important observation because it allows us to isolate a small functional block which is repeated throughout the calculation circuitry.

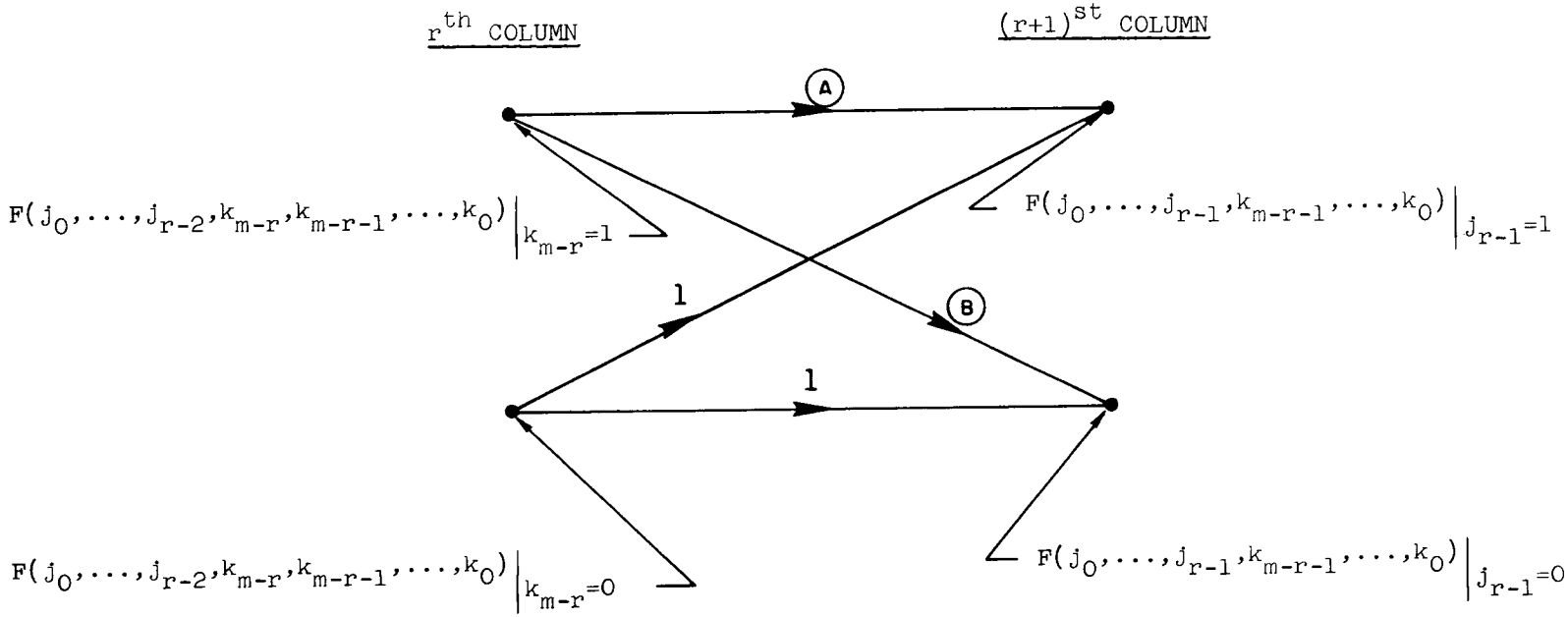
This observation also sets a lower limit on the number of operations that must be performed in parallel; that is, we must calculate the points in pairs. After each pair of the $(r+1)^{st}$ column is calculated, the corresponding points of the r^{th} column may be discarded and calculation of another pair started. Hence, we must, as a minimum, provide storage for $n+2$ complex numbers.

Figure 19 illustrates the flow chart of this basic functional block described in the previous paragraph. We may split the basic operation in half so that we will be calculating, for every point, the sum of one previous point and one point multiplied by a constant according to Eq. (6.1).

$$
\begin{aligned}
&F(j_0,\ldots,j_{r-1},k_{m-r-1},\ldots,k_0) \\
&\quad = F(j_0,\ldots,j_{r-2},k_{m-r},k_{m-r-1},\ldots,k_0)\Big|_{k_{m-r}=0} \\
&\quad + F(j_0,\ldots,j_{r-2},k_{m-r},k_{m-r-1},\ldots,k_0)\Big|_{k_{m-r}=1} \\
&\qquad \exp\left[-i2\pi\left(j_{r-1}2^{-1} + j_{r-2}2^{-2} + \ldots + j_0 2^{-r}\right)\right]
\end{aligned}
\tag{6.1}
$$

The value of the exponential constant is dependent only on where we are in the overall calculation flow chart. The fact that it is a constant will make the multiplying circuitry simpler.

Since the multiplying constant has a magnitude of less than one, as we move from one column to the next, the



(A) $\exp\left[-i2\pi\left(2^{-1} + j_{r-2}2^{-2} + j_{r-3}2^{-3} + \ldots + j_0 2^{-r}\right)\right]$

(B) $\exp\left[-i2\pi\left(j_{r-2}2^{-2} + j_{r-3}2^{-3} + \ldots + j_0 2^{-r}\right)\right]$

Fig. 19. FLOW CHART OF BASIC COMPUTATIONAL BLOCK.

magnitude of the numbers will increase by less than a factor of 2. Hence, as we move to the right each column will in general require one more bit in its representation than did the previous column. Consequently, if we begin with 2^m each q-bit samples, the spectrum calculated will contain 2^m words of $q+m$ bits each.

B. Basic Functional Block Using Only One Multiplier

The relationship between the two multiplicative constants needed can be seen in the basic functional block of Fig. 19. Note specifically that the constant on the top path can be written

$$\exp\left[-i2\pi\left(2^{-1} + j_{r-2}2^{-2} + j_{r-3}2^{-3} + \ldots + j_0 2^{-r}\right)\right]$$

$$= \exp(-i\pi)\exp\left[-i2\pi\left(j_{r-2}2^{-2} + j_{r-3}2^{-3} + \ldots + j_0 2^{-r}\right)\right]$$

$$= -\exp\left[-i2\pi\left(j_{r-2}2^{-2} + j_{r-3}2^{-3} + \ldots + j_0 2^{-r}\right)\right] \quad (6.2)$$

The right-hand side of (6.2) is seen to be just the negative of the multiplicative constant on the downward-sloping path. Hence Fig. 19 can be modified so that only one complex multiplication is needed rather than the two required by straightforward application of the flow graph. Figure 20 shows the modified basic functional block.

This modification reduces the total number of complex multiplications for the complete flow chart from $m2^m$ to $m2^{m-1}$. Since the majority of the computation time and

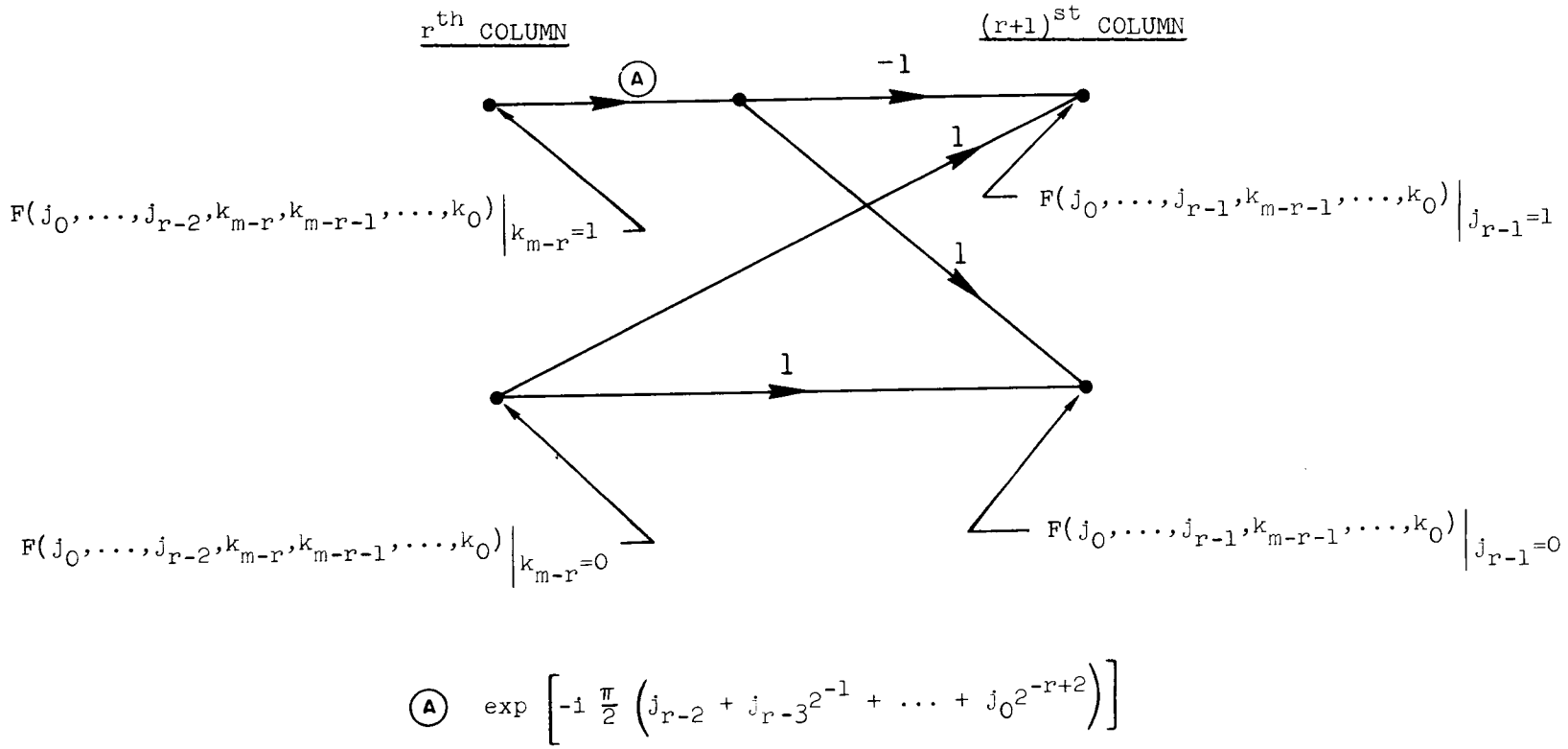


Fig. 20. MODIFIED FLOW CHART OF BASIC COMPUTATIONAL BLOCK WHICH REQUIRES ONLY ONE COMPLEX MULTIPLICATION.

circuitry is spent in multiplication, this simple modification allows a savings of about one-half of the circuitry necessary.

C. Hardware Adders

The operations of addition and multiplication must be implemented. A parallel binary adder of the appropriate number of bits can be quite easily implemented using currently available integrated circuits. Figure 21 illustrates a 12-bit adder constructed from 4-bit adder integrated circuits which are available commercially from Texas Instruments.

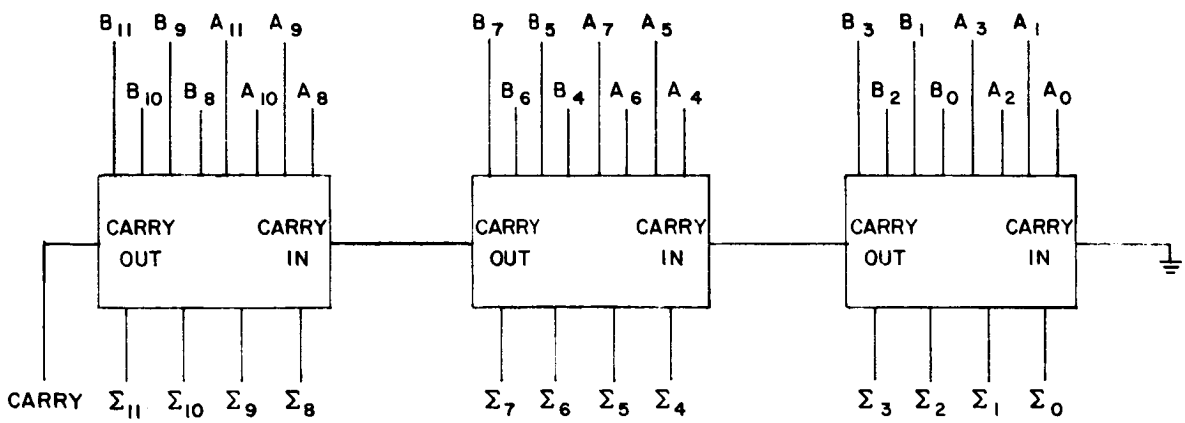


Fig. 21. A 12-BIT STRAIGHT BINARY ADDER. All integrated circuits are Texas Instruments 4-bit adder modules.

D. Hardware Multipliers

Multiplication of two binary numbers can be accomplished by a shifting and adding procedure. When one of the numbers is known ahead of time (as are the exponential constants), the hardware may be greatly simplified. The simplification,

however, makes one multiplier different from another and hence increases the number of types of circuits needed.

Figure 22 shows one possible implementation of a 12-bit straight binary multiplier. This is a completely general

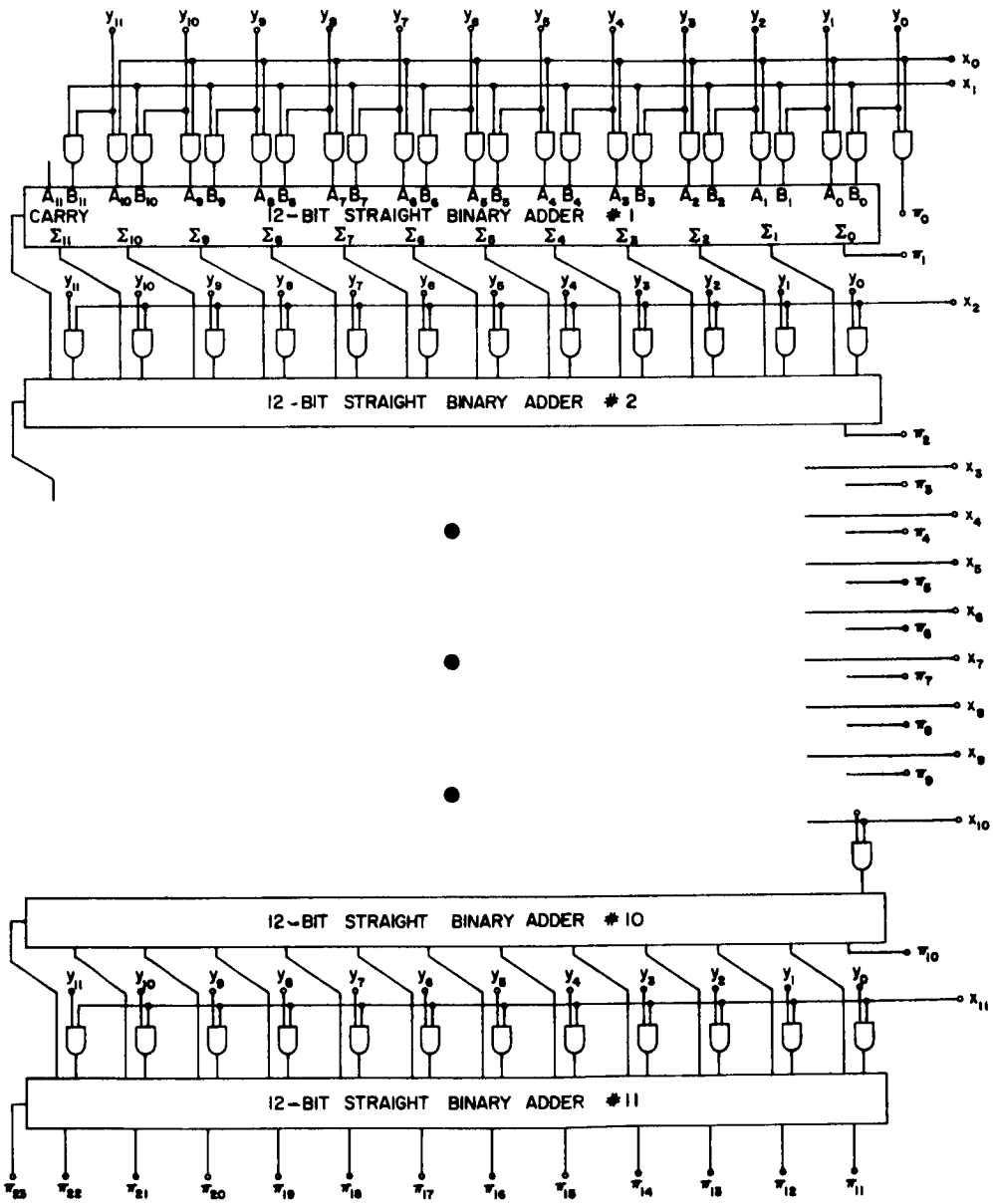


Fig. 22. A 12-BIT STRAIGHT BINARY MULTIPLIER.

circuit since each of the multiplicands x and y can be any binary number. The building of a multiplier in this fashion has in general not been economically feasible in the past, and actually probably is not yet. However, as integrated circuit technology progresses, this multiplier becomes more and more attractive. For instance, at the present time Texas Instruments is producing commercially a 4-bit parallel adder in a single integrated circuit chip. At the present state of the art, then, the multiplier of Fig. 22 would require something on the order of 72 integrated circuit chips to implement with commercially available products. This number could probably be cut in half through design of special-purpose chips. It is not inconceivable that this complete circuit may some day be implemented in a single chip.

E. Simplification of Multipliers

When the value of one of the multiplicands is known ahead of time, the multiplier of Fig. 22 may be greatly simplified. Knowing the values of x_0 through x_{11} will allow us to eliminate all the AND gates and, on the average, half the adders from Fig. 22. To prove this, note that if any x_i is 1 then we may eliminate the associated row of AND gates and simply tie the y_i directly into the associated adder. Further, if any x_i is 0 we may eliminate both the associated AND gates and the associated adder since the sum of the adder is simply equal to the other input. The x_i

will on the average be half 1's and half 0's; hence on the average we may eliminate half the adders.

Thus the necessary multipliers could be made using an average of about 15-1/2 of the existing 4-bit adder chips currently available from Texas Instruments. This is a considerable savings over the previously mentioned 72 chips per multiplier.

Since the numbers dealt with will be complex, every point of the calculation will require four multiplications and two additions. If there are 2^m sample points, there will be $4m2^m$ multiplications and $2m2^m$ additions. If 12-bit numbers are used as proposed, the results of the multiplications will be 24-bit numbers which must be rounded off again to 12 bits before we add. This will usually involve merely dropping the 12 least significant bits. The multipliers can be simplified still further if these bits are not calculated at all, since they will be dropped anyway.

Chapter VII

SPEED OF OPERATION

A. Calculation Time

The time required to perform the calculations will set an upper limit on the sample rate and thus set an upper limit on frequency response of our spectrum analyzer. It will be instructive to work out an example to see what can be done with currently available products. The Texas Instruments 4-bit adder has a carry propagation delay of 30 ns for 4 bits or 90 ns for 12 bits. This is the longest path for the 12-bit adder. The longest path for the multiplier is through the 90 ns of carry delay and down the average of 5-1/2 rows of adders at about 50 ns each, then through another 90 ns of carry delay for a total of 445 ns. Add to this the 90 ns of the addition and we get about 545 ns for each column of the calculation.

Using this value it is possible to derive Table 1, which relates the number of 12-bit samples used to the approximate maximum sampling rate and consequently the highest frequency estimate of the resulting digital spectrum analyzer. Note that the calculation time varies with the number of columns or the log of the number of samples. The calculation time also increases roughly directly proportional to the number of bits in the samples.

Table 1

NUMBER OF SAMPLES VS CALCULATION TIME AND MAXIMUM SAMPLE RATE FOR 12-BIT SAMPLE WORDS (0.025 PERCENT ACCURACY)

m	Number of Samples (2^m)	Calculation Time (μs)	Maximum Sample Rate (kHz)	Maximum Frequency Estimate (kHz)
1	2	0.545	1840	920
2	4	1.09	918	459
3	8	1.635	612	306
4	16	2.18	458	229
5	32	2.73	367	183
6	64	3.27	306	153
7	128	3.81	262	131
8	256	4.36	229	115
9	512	4.91	204	102
10	1,024	5.45	184	92
11	2,048	6.00	167	84
12	4,096	6.54	153	77
13	8,192	7.08	141	71
14	16,384	7.63	131	66

Referring to Table 1, it should be noted that the maximum sample rates are somewhat faster than the conversion rates of currently available 12-bit A/D (analog to digital) converters whose conversion times will be in the range of 20 to 100 μs. This need not be a problem (except possibly a financial one) since it should be an easy matter to use several "Sample and Hold" amplifiers followed by several A/D converters, and then use a digital multiplexer to present the digitized samples to the calculator in the correct time sequence.

The described implementation of a digital spectrum analyzer is several orders of magnitude faster than other schemes realized on digital computers (e.g., Weaver et al [1966]; Cooley & Tukey [1965]; Larson & Singleton [1967]; Gentleman & Sande [1966]). It is this extra speed that should make possible a practical real-time digital spectrum analyzer. Whether or not the extra cost of a specially constructed parallel arithmetic unit is "practical" will depend on the particular application.

B. Pipelining

Some increases in speed may be made through use of a technique known as "pipelining." In this technique, a second set of samples is input into the calculator before the results of the first set of data have come out the output end. In this way the data sets will proceed in waves from input to output in the calculator.

In general, the number of waves is limited by the differences in time delays along different paths in the calculation. That is, the results will be meaningful only so long as one wave does not overtake the wave in front of it or lag into the wave behind it. Otherwise the results will become scrambled and meaningless.

If all paths in the calculation required exactly the same time, the calculation could be pipelined up to any desired speed. Unfortunately it is a practical impossibility to make all the paths the same time length. If, however, a stage of

memory is inserted between each column of the calculation, we can gain in speed by a factor of m for 2^m samples. Figure 23 shows how this would be accomplished.

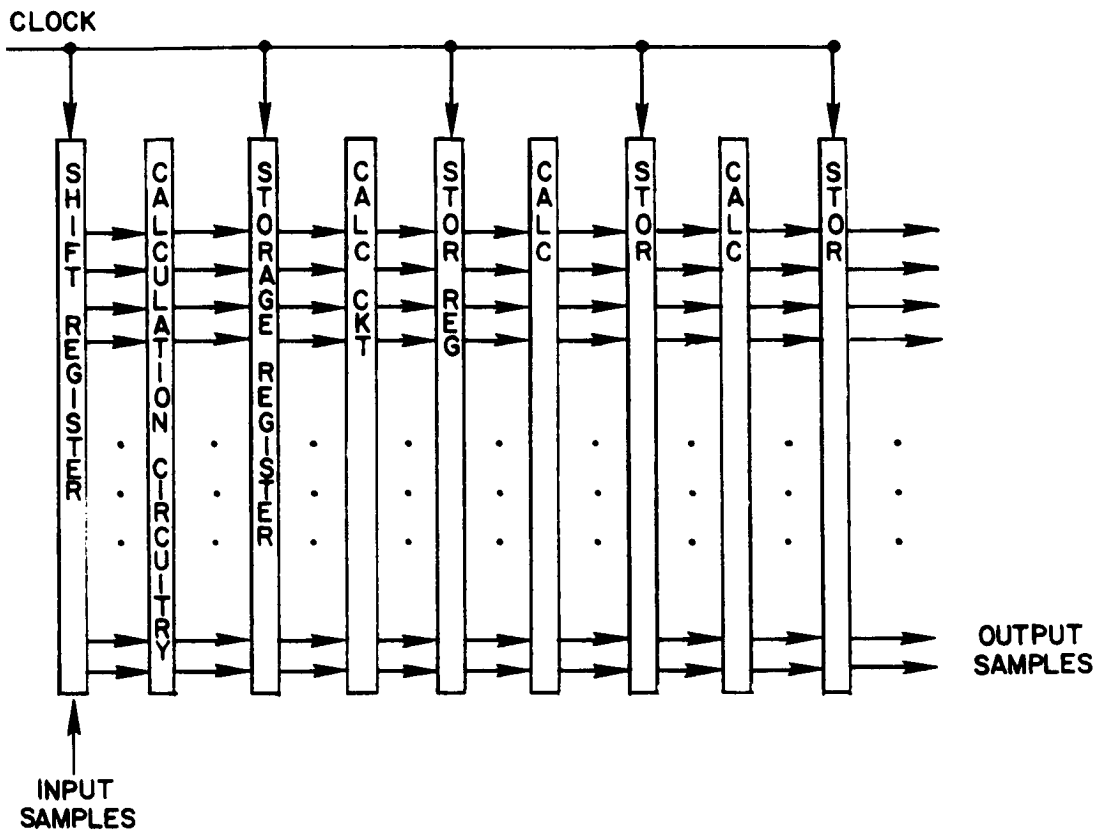


Fig. 23. CALCULATION CIRCUITRY UTILIZING PIPELINING.

By clocking the storage registers and the input shift register we can essentially make the delay along all paths of the calculation constant. The delay will be equal to one clock period times the number of columns in the calculation. The clock period need only be long enough to allow for the maximum time delay through a single column of the calculator. This maximum delay would be about 820 ns for the multiplier

and adder described earlier. Thus the clock could be run at a 1.2 MHz rate regardless of the number of input samples used. This in turn means a sample rate and output rate of 1.2 MHz. The maximum frequency estimate then would be raised to 600 kHz. And all this in real time!

The higher sampling rate, of course, means that even more sample and hold circuits, more A/D converters, and more inputs to the digital multiplexer would be needed. Thus, while the potential for high-speed operation exists, it is very expensive to realize.

Chapter VIII

REAL-TIME ALGORITHM

A. Real-Time Operation

When operating in real time we will run the input through an analog-to-digital converter and store the results in a shift register to be used as an input to the calculation circuitry. As each new sample is brought in, all samples in the register will be shifted down one position and the frequency estimates will be recalculated.

When operating in this manner, the algorithm as described in the previous chapters performs redundant calculations. By modifying the algorithm to remove these redundancies in real-time operation, it is possible to decrease the amount of hardware necessary by approximately a factor of $1/(\log_2 n)$ for n input samples.

B. Derivation of Real-Time Algorithm

Figure 24 illustrates the modified algorithm. The operations indicated in dashed lines between the first two columns are among those that have been eliminated. Since the input samples are being shifted one place each sample time, at any time t the dashed operations would be calculating exactly the same results that the solid operations calculated during the previous sample time $t - \tau$. Consequently, rather than repeat the calculation we may shift the results of the previous calculation into this register. Note that this is

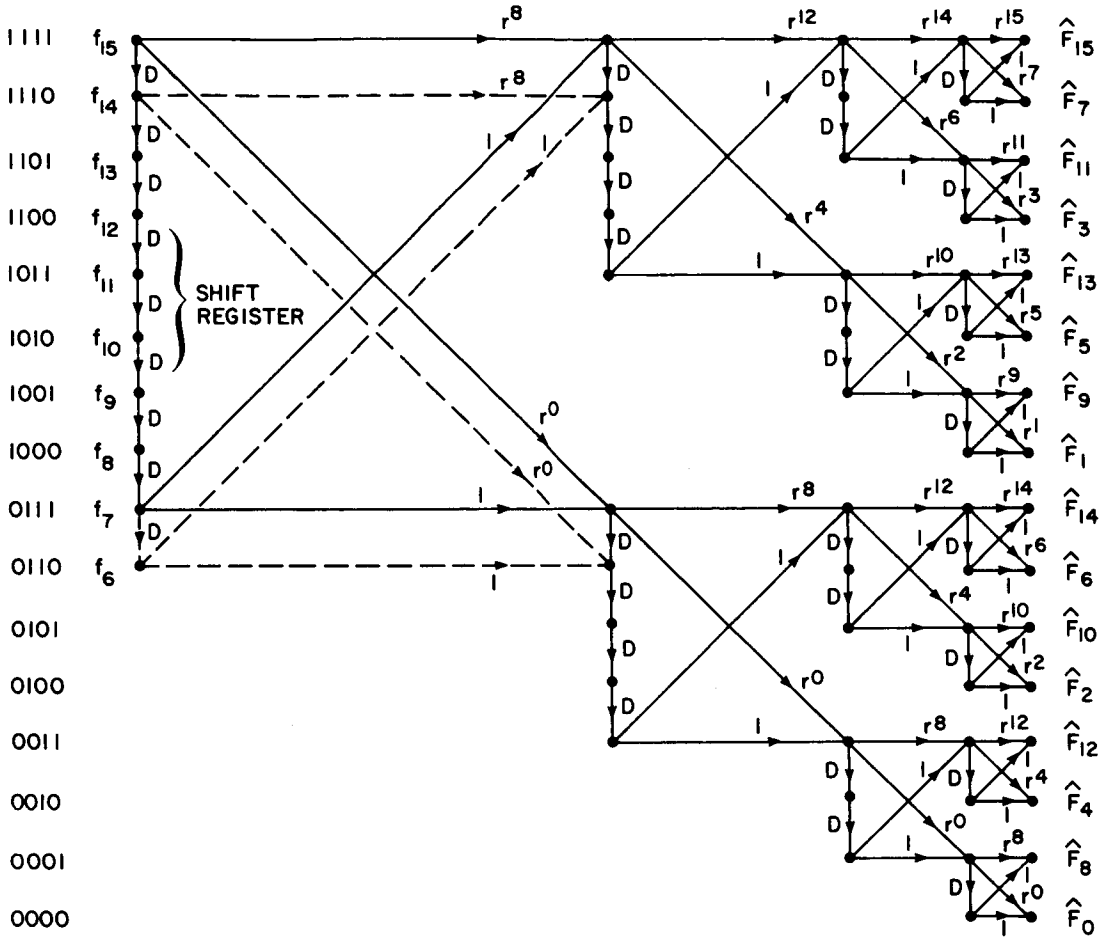


Fig. 24. MODIFIED ALGORITHM FOR REAL-TIME OPERATION.

possible only because the constant multipliers were the same for both the solid and the dashed operations. Further study of the flow chart of Fig. 5 will indicate that the complete second column of this figure may be generated by shifting downward the results in row 7 and row 15.

Similarly the results in the third column may be generated by shifting downward the results in rows 3, 7, 11, and 15. Continuing in this fashion through all the columns, we

arrive at the much simplified algorithm shown in Fig. 24. By using the single multiplier basic function block of Fig. 20, the flow graph can be simplified further as shown in Fig. 25.

C. Hardware Requirements

Figure 25 requires one complex multiplier between columns 1 and 2; two between columns 2 and 3; four between columns 3 and 4; and so forth up to 2^{m-1} between the last two columns. The total number N of complex multipliers needed is

$$N = 1 + 2 + 4 + 8 + \ldots + 2^{m-1} \tag{8.1}$$

This series can be brought into closed form as follows:

$$2N = 2 + 4 + 8 + \ldots + 2^{m-1} + 2^{m} \tag{8.2}$$

$$1 + 2N - 2^{m} = 1 + 2 + 4 + 8 + \ldots + 2^{m-1}$$

$$= N \tag{8.3}$$

Collecting terms yields $N = 2^{m} - 1$ complex multipliers, which amounts to a considerable savings in hardware over a direct parallel implementation of the Cooley-Tukey algorithm which would require $N = m2^{m}$ complex multipliers.

The real-time algorithm of Fig. 25 can also be very easily pipelined as mentioned in Chapter VII. In this case, then, the shift registers would not represent an additional storage register requirement but would merely be used in place of the clocked registers shown in Fig. 23. The real-time algorithm,

in fact, uses fewer registers than the original. Figure 23 will require $(m+1)$ columns of 2^m registers each for $(m+1)2^m$ registers.

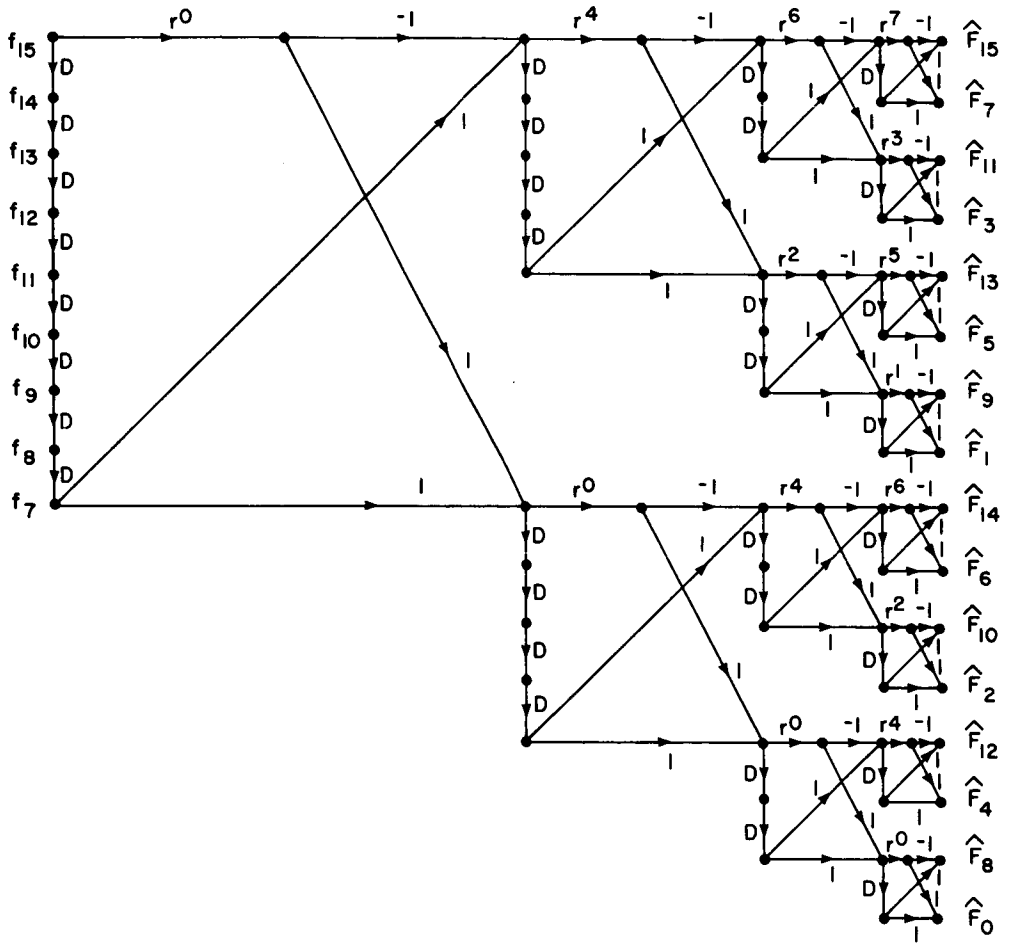


Fig. 25. REAL-TIME ALGORITHM USING ONLY ONE MULTIPLIER PER BASIC FUNCTION BLOCK.

The real-time algorithm will require $2^{m-1} + 1$ registers in the first column; $2^{m-1} + 2$ registers in the second; $2^{m-1} + 4$ in the third; and so on up to 2^m in the m^{th} and $(m+1)^{st}$ columns. The total number R of registers needed is given by

$$R = (m+1)\ 2^{m-1} + 1 + 2 + 4 + \ldots + 2^{m-2} + 2^{m-1} + 2^{m-1}$$

$$= (m+1)\ 2^{m-1} + 2^{m} - 1 + 2^{m-1}$$

$$= 2^{m}\left(\frac{1}{2}\, m + 2\right) - 1 \qquad (8.4)$$

which is less than the $m2^m$ needed for straightforward pipelining of the Cooley-Tukey algorithm for all $m \geq 4$.

There is yet another great advantage of the real-time algorithm that may not be immediately obvious. Specifically, the advantage lies in the fact that only the first and last stages of each shift register are looked at. This means that cheaper types of shift registers may be used. For instance, a simple delay line could be used as a shift register since only the input and output are used. The delay line unfortunately makes it virtually impossible to change sampling rates. Although the delay line offers a large amount of storage capacity at low cost, the problems of synchronizing its operation with various sampling rates preclude its further consideration.

In the area of clocked shift registers, there are several choices. For example, a magnetic-core shift register can be used since we don't have to look at all the intermediate stages. The data in the first and last stages would be held in flip-flop registers so that it would be available as inputs to the multipliers and adders. The magnetic-core shift

register is more easily adapted to changing sample rates than is the delay line.

Another possibility is the integrated circuit (IC) shift register, such as Philco's 100-bit MOS shift register in a single TO-5 can. Use of this device is possible only because the real-time algorithm does not require access to the intermediate stages along the shift register string.

It is important to note that if it were necessary to have access to the intermediate stages, the size of the IC package would have to be almost 100 times larger! The size of these packages is determined primarily by the number of inputs and outputs necessary for the circuit. The IC chip itself is usually far smaller than the package containing it.

The MOS shift register unfortunately has not only a 1.5 MHz maximum clock rate, but also a 5 kHz minimum clock rate. The clock-rate limitation stems from the fact that the device employs MOS gate capacitance for temporary storage and consequently will not operate at dc. Here again we would be sample-rate limited.

Perhaps the best commercial integrated circuit for this application at the present time would be an 8-bit T^2L shift register made by Texas Instruments. It is a serial-in, serial-out unit that will operate from dc to 18 MHz, and consequently will not be sample-rate limited. Also all shift registers in the real-time algorithm of eight stages or more can be divided evenly by 8.

Future IC developments will undoubtedly produce longer IC shift registers. Serial-in, serial-out units will be developed first since they can be encased in current IC packages. They are not subject to the same pin limitation problems that large parallel-in, parallel-out shift registers are. Since more can be squeezed into the same package, both parts costs and wiring costs will be much less with the real-time algorithm.

Chapter IX

SINGLE AND MULTIPLE CHANNEL FILTERS

A. Realization

The real-time algorithm may be used to realize single or multiple channel filters up to n channels. These filters may be realized simply by eliminating from the real-time algorithm all the operations which do not lead to the frequency estimates desired. The procedure is best illustrated by an example. Figure 26 shows the derivation of a filter for estimating $\hat{F}_{10}$ from the flow chart of the real-time algorithm. The dashed lines indicate portions of the flow graph which can be eliminated since they are isolated from the $\hat{F}_{10}$ output.

It is obvious from this figure that any output or any combination of outputs can be chosen. Two of these single channel filters--$\hat{F}_0$ and $\hat{F}_8$-- warrant special attention because of their ease of implementation and their other special features. Figure 27 shows the flow chart for $\hat{F}_0$, which is the estimate of the dc content of the input signal. Notice that the only multiplier used is r^0. But $r^0 = 1$! This means that $\hat{F}_0$ can be calculated using no multipliers at all--only delay lines and adders are needed. Also, assuming the input is real, all numbers in the calculations are real rather than complex. The lack of multipliers and complex numbers results in tremendous hardware savings in the $\hat{F}_0$ lowpass filter.

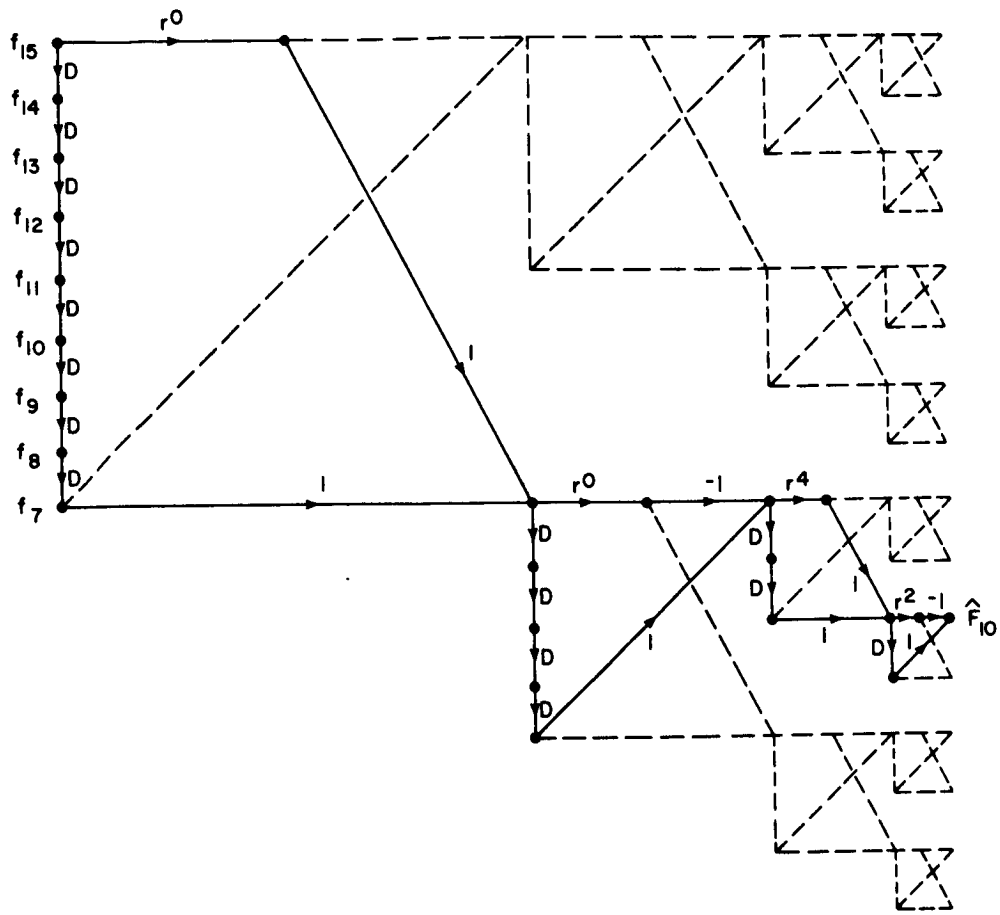
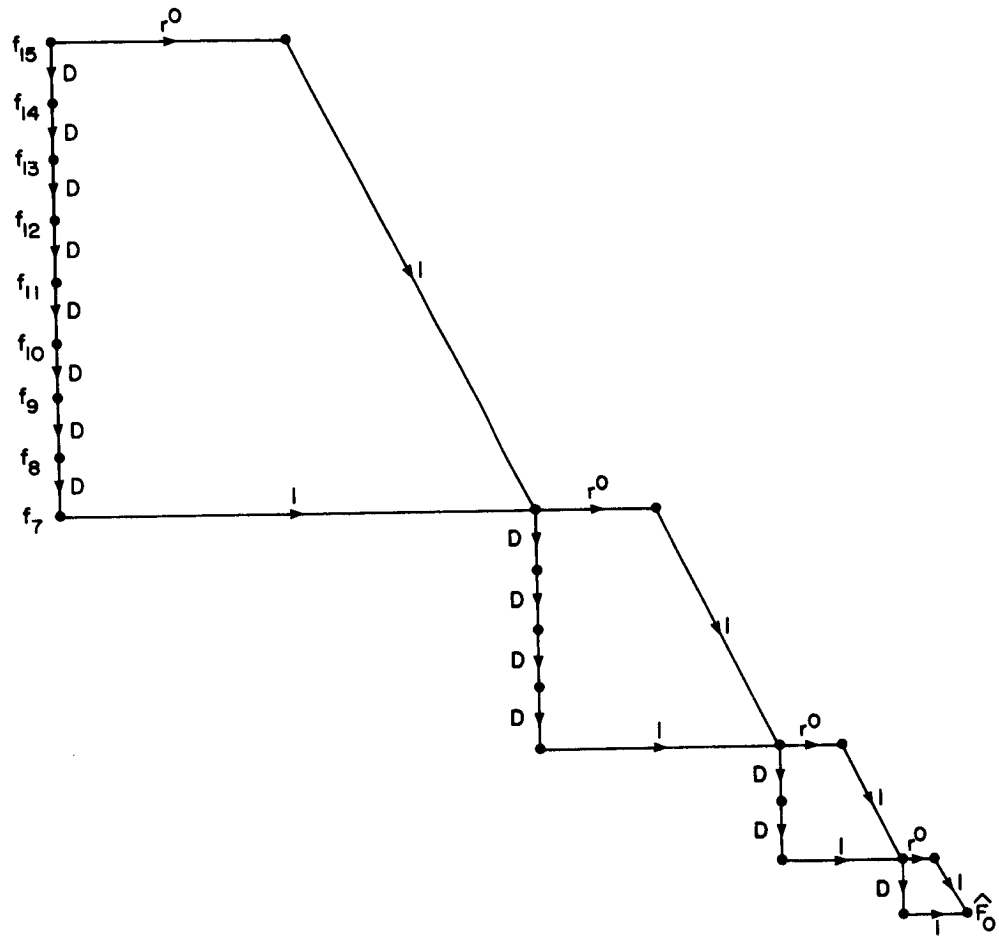


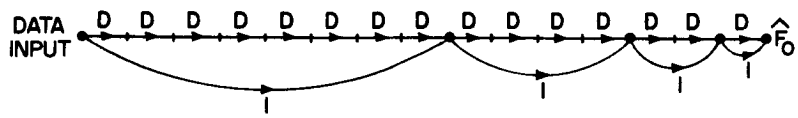
Fig. 26. DERIVATION OF SINGLE-CHANNEL FILTER FOR $\hat{F}_{10}$ FROM REAL-TIME ALGORITHM.

Since the spectrum is "folded" back on itself by the sampling, the highest frequency estimate is $\hat{F}_8$ rather than $\hat{F}_{15}$. Figure 28 shows the flow chart for $\hat{F}_8$. As with $\hat{F}_0$, the only multiplication is by r^0, or 1. Hence the hardware for calculating $\hat{F}_8$ also requires no multipliers or complex numbers. Another advantage of the single channel filters is that the resolution is readily increased simply by adding on

another stage of shift register, multiplier, and adder at the input end.



a. Filter derived from real-time algorithm

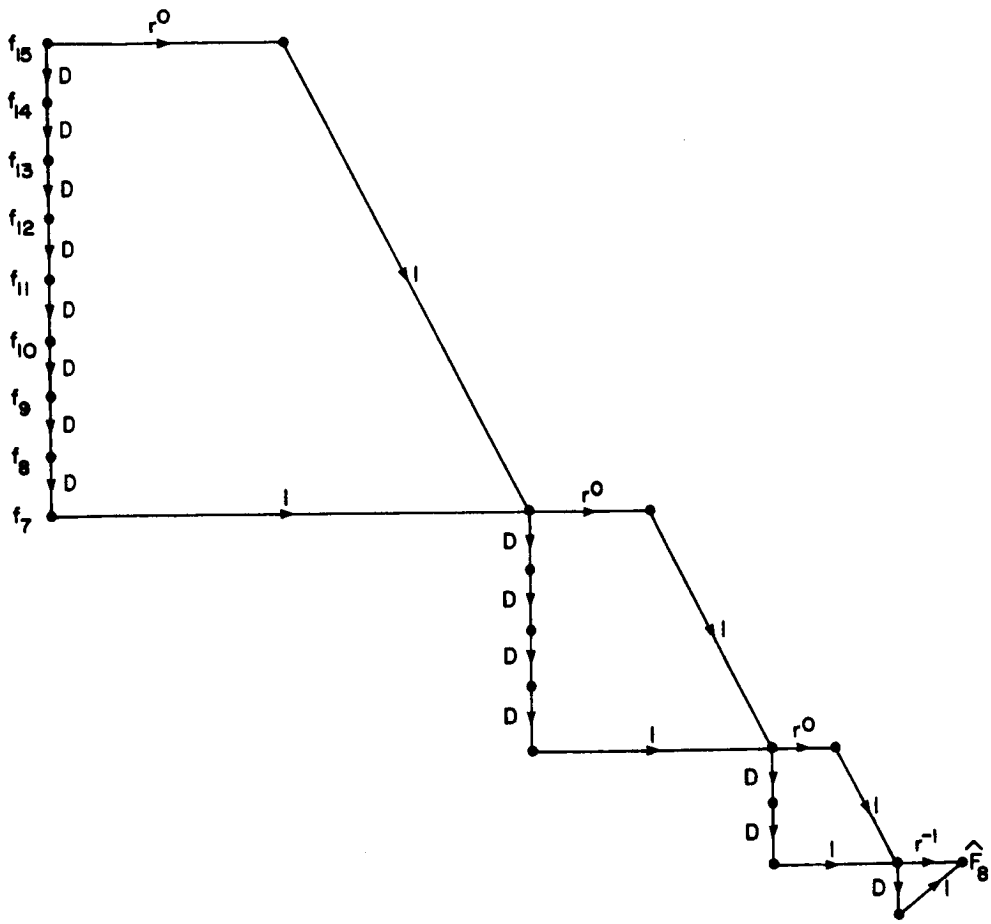


b. Filter simplified

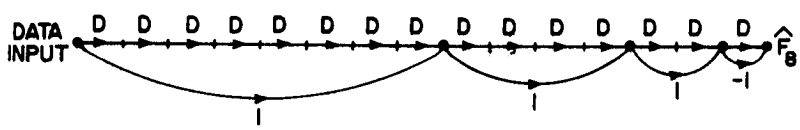
Fig. 27. DIGITAL LOWPASS FILTER DERIVATION.

B. Applications

As an example of the possible use of the bandpass filter for $\hat{F}_8$ we cite the following: Satellite-tracking ground



a. Filter derived from real-time algorithm



b. Filter simplified.

Fig. 28. $\hat{F}_8$ PASSBAND FILTER DERIVATION.

stations as a rule derive their time information from a central time reference source. The information is usually sent out as a PDM (pulse duration modulation) waveform which is used to amplitude-modulate a 1 kHz carrier. By setting up the filter such that $\hat{F}_8$ occurs at 1 kHz, it should be

possible to pick off the information in the modulating envelope.

In an application such as this, a fixed sampling rate would be used and costs could be cut further by using a delay line in place of the shift register. A clocked register would be used at the output of the delay line to furnish the input to the adder. Thus the delay time of the line could vary up to one clock period without causing an error in the output.

Filters such as these may also find application in analyzing data containing very low frequencies. Analog filters become very large physically at frequencies of, say, 1 Hz and below. Such frequencies are important in the areas of earthquakes and medicine. In medicine the EEG (electroencephalogram) and the EKG (electrocardiogram) would be prime examples of areas where frequencies below 1 Hz occur. Consequently they are also areas in which digital filters such as described above should be useful.

Chapter X

DIGITAL SPECTRUM ANALYZER OUTPUT

A. Complex Sinusoidal Output

Up to this point we have spoken of the analyzer output as though it were a constant. The fact is, of course, that each output of the analyzer is a time series which represents a running estimate of the spectral content of the input based on the last n samples.

The time axis of the Fourier transform is fixed to the transform circuitry and consequently moves with respect to the input time waveform. Figure 29 illustrates the shifting of the input past the data window of the transform circuitry.

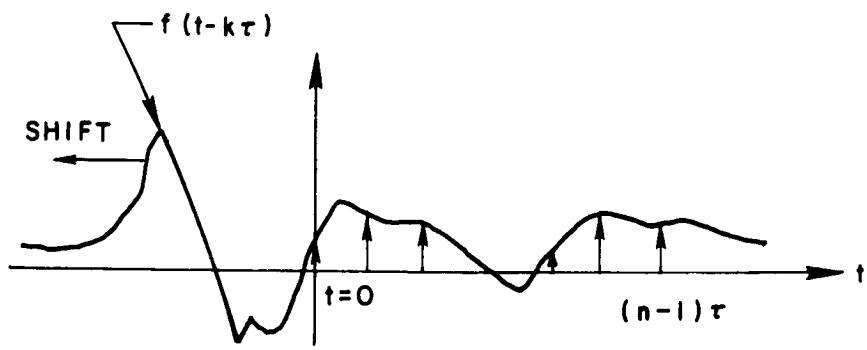


Fig. 29. SHIFTING OF $f(t)$ PAST TRANSFORM CIRCUITRY.

At time $t = (n-1)\tau$ the spectrum estimates are based on the input samples at $t = 0$ through $t = (n-1)\tau$. Rather than

calling this estimate $\hat{F}(\omega_0)$ as we did before, let us now refer to it as $\hat{F}(\omega_0,\ t = (n-1)\tau)$.

At some later time $t = (n-1)\tau + k\tau$, the inputs used will be $f(k\tau)$ through $f[(n-1)\tau + k\tau]$. The transform of the shifted input $f(t - k\tau)$ and the transform of the sampling function are shown in Fig. 30. The multiplication in the time domain results in a convolution in the frequency domain. The resulting spectrum can be expressed as

$$\hat{F}(\omega_0,\ t = (n-1)\tau + k\tau) = \frac{n}{8\pi^3} \int_{-\infty}^{\infty} F(\omega)\ \exp(-ik\tau\omega)$$

$$\cdot \exp\left[-i\ \frac{n\tau}{2}\ (\omega_0 - \omega)\right]\ \frac{\sin\ \frac{n\tau}{2}\ (\omega_0 - \omega)}{\frac{n\tau}{2}\ (\omega_0 - \omega)}\ d\omega \qquad (10.1)$$

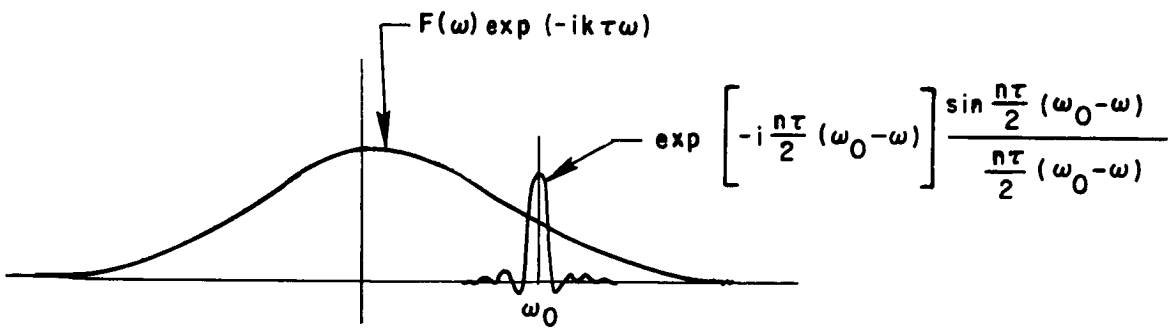


Fig. 30. DERIVATION OF $\hat{F}(\omega_0,\ t = (n-1)\tau + k\tau)$.

In the vicinity of ω_0, this can be approximated by

$$\hat{F}(\omega_0,t) \approx \frac{n}{8\pi^3} \int_{-\infty}^{\infty} F(\omega_0) \exp(-ik\tau\omega_0) \cdot \exp\left[-i \frac{n\tau}{2} (\omega_0 - \omega)\right] \frac{\sin \frac{n\tau}{2} (\omega_0 - \omega)}{\frac{n\tau}{2} (\omega_0 - \omega)} d\omega \tag{10.2}$$

The terms that don't depend on ω can be brought outside the integral.

$$\hat{F}(\omega_0,t) \approx \frac{n}{8\pi^3} F(\omega_0) \exp(-ik\tau\omega_0) \int_{-\infty}^{\infty} \exp\left[-i \frac{n\tau}{2} (\omega_0 - \omega)\right] \frac{\sin \frac{n\tau}{2} (\omega_0 - \omega)}{\frac{n\tau}{2} (\omega_0 - \omega)} d\omega \tag{10.3}$$

Carrying out the integration yields

$$\hat{F}(\omega_0,t) \approx \frac{F(\omega_0) \exp\left\{-i\omega_0[t - (n-1)\tau]\right\}}{8\pi^2\tau} \tag{10.4}$$

It should be remembered that Eq. (10.4) was derived by assuming that $F(\omega) \exp(-i\omega k\tau)$ was constant throughout the passband at ω_0. Under this assumption, then, the output of the filter is a real and an imaginary sinusoid, both of frequency ω_0. The information we are seeking, namely $F(\omega_0)$, is contained in the envelopes of the sine waves.

$F(\omega_0)$ itself is a complex number. If we set $F(\omega_0) = F_r + iF_i$, we may write

$$\hat{F}[\omega_0,\ t = (n-1)\tau + k\tau] \approx \frac{1}{8\pi^2\tau}\left(F_r \cos k\tau\omega_0 + F_i \sin k\tau\omega_0\right) + \frac{i}{8\pi^2\tau}\left(-F_r \sin k\tau\omega_0 + F_i \cos k\tau\omega_0\right) \tag{10.5}$$

Let us now take the magnitude squared of (10.5) for an estimate of the power spectral density at ω_0.

$$|\hat{F}[\omega_0,\ t = (n-1)\tau + k\tau]|^2 \approx \frac{1}{64\pi^4\tau^2}\left[F_r^2 \cos^2(k\tau\omega_0) + 2F_rF_i \cos(k\tau\omega_0)\sin(k\tau\omega_0) + F_i^2 \sin^2 k\tau\omega_0\right] + \frac{1}{64\pi^4\tau^2}\left(F_r^2 \sin^2 k\tau\omega_0 - 2F_rF_i \cos k\tau\omega_0 \sin k\tau\omega_0 + F_i^2 \cos^2 k\tau\omega_0\right) \tag{10.6}$$

Making use of the $\sin^2 x + \cos^2 x = 1$, and by collecting terms, we can write

$$|\hat{F}[\omega_0,\ t = (n-1)\tau + k\tau]|^2 \approx \frac{1}{64\pi^4\tau^2}\left(F_r^2 + F_i^2\right) \approx \frac{1}{64\pi^4\tau^2}|F(\omega_0)|^2 \tag{10.7}$$

Note that the time dependency has dropped out of the right-hand side of Eq. (10.7). Thus, while the voltage transform outputs vary sinusoidally with time, the power spectrum estimate is approximately constant with time. The rapid change of the voltage transform should be expected since the voltage transform contains information about the phase of the input relative to the time reference of the transform circuitry. Since the input is shifted past the transform circuitry, its phase relative to the transform time reference changes rapidly.

B. Updating Frequency Estimates

When we calculate the spectral estimates, we always do so on the basis of the last n input samples. When we recalculate the estimates one sample period later, the calculations are based on $n-1$ of the original samples. That is, we drop the oldest sample and add one new one and recalculate. Since the majority of the calculation is based on the old samples, it would seem that we are generating redundant information.

Can we not, in fact, generate the second set of estimates simply by applying Eq. (10.4), with $k = 1$, to the first set of estimates? The answer is that this does indeed come pretty close to predicting the second set of estimates. We may apply (10.4) again to predict the third set of estimates; and again to predict the fourth set; and so forth. As might be expected, the further we predict, the greater the probable errors become.

The source of the errors is our approximation that $F(\omega)\exp(-ik\tau\omega)$ is essentially constant through the passband of the filter. For $k = 1$ (predicting one sample period ahead), the approximation is relatively good. As we increase k (predicting further and further ahead), the term $\exp(-ik\tau\omega)$ varies more and more rapidly with ω. Consequently our approximation (and our predictions) get worse and worse.

C. Frequency Components of Envelope

Referring back to Eq. (10.1) we see that $\hat{F}(\omega_0,t)$ is the output of a narrowband filter. Consequently it will contain frequency components between $\omega - (\Delta\omega/2)$ and $\omega_0 + (\Delta\omega/2)$ where $\Delta\omega$ is the bandwidth of the filter. Equation (10.4) tells us that $\hat{F}(\omega_0,t)$ can also be written approximately as an envelope $F(\omega_0)/8\pi^2\tau$ times $\exp(-ik\tau\omega_0)$. If we assume that the envelope is slowly varying sinusoidally at a frequency θ, we can write

$$\frac{F(\omega_0)}{8\pi^2\tau} \approx Ae^{ik\tau\theta} + Be^{-ik\tau\theta} \qquad (10.8)$$

The product of the envelope and the sinusoid can be written:

$$\frac{F(\omega_0)}{8\pi^2\tau} e^{-ik\tau\omega_0} \approx Ae^{-ik\tau\omega_0} e^{ik\tau\theta} + Be^{-ik\tau\theta} e^{-ik\tau\omega_0} \qquad (10.9)$$

This can be compressed to

$$\frac{F(\omega_0)}{8\pi^2\tau} e^{-ik\tau\omega_0} \approx Ae^{-ik\tau(\omega_0-\theta)} + Be^{-ik\tau(\omega_0+\theta)} \qquad (10.10)$$

Thus the product of a complex sinusoidal envelope and complex exponential produces frequency components displaced $\pm\theta$ from the ω_0 frequency of the exponential. Since no frequency components of $\hat{F}(\omega_0,t)$ are further than $\pm\Delta\omega/2$ away from the center frequency ω_0, then the envelope $F(\omega_0)/8\pi^2\tau$ must contain no frequency components higher than $\Delta\omega/2$. This corresponds to a period of $4\pi/\Delta\omega$.

We can theoretically regain the envelope alone by multiplying the output of the filter by $\exp(ik\tau\omega_0)$. Having done this, the sampling theorem tells us we may extract all the information in the envelope by sampling it twice every period--i.e., at a time spacing of $2\pi/\Delta\omega$.

We defined $\Delta\omega$ above as the bandwidth of the analyzer channel outside of which there is zero transmission of frequency components. By referring to the calculated passband characteristics of Figs. 12, 13, and 14, and even the double-hanning passband of Fig. 18, we can see that the passband never falls off completely to zero. Consequently we cannot put an absolute frequency limit on the envelope. However, we can say that the frequency components beyond a certain point become negligibly small.

If we pick as our filter bandwidth the first zero crossing of the $(\sin x)/x$ passband characteristic of Fig. 12, then the bandwidth is given by

$$\Delta\omega = \frac{4\pi}{n\tau} \tag{10.11}$$

Consequently the output envelope must be sampled at a period of

$$\frac{2\pi}{\Delta\omega} = \frac{n\tau}{2} \tag{10.12}$$

It should be remembered that (10.12) was derived by <u>not</u> using the more standard half-power point definition of bandwidth. By using this definition, we obtain approximately:

$$\frac{2\pi}{\Delta\omega} \approx n\tau \tag{10.13}$$

Equation (10.13) states, in effect, that we need not look at all the analyzer outputs after every input sample, but rather, only once every n input samples.

Chapter XI

TIME-SHARED REAL-TIME ALGORITHM

A. Derivation of Algorithm

In Chapter X we found that the output of the Fourier transform circuitry can be expressed as a complex sinusoid with a slowly varying envelope. In order to reproduce the envelope which carries information about $F(\omega)$, we need look at the outputs only once every n inputs. We are consequently led to inquire whether or not we can reduce the hardware requirements by calculating the outputs in rotation. By this we mean calculating a different one of the outputs each sample time so that in n sample times we have calculated all n outputs. This process would then be repeated. Each output calculated would be based on the set of the immediately preceding n inputs, as before. Now, however, since the outputs are not calculated at the same time, they do not depend on the same sample sets. Figure 31 shows a possible time sequence of outputs and the input sample sets upon which they are based.

Let us refer back to Fig. 25. Note that in order to calculate any one of the points in the output column, we must calculate two consecutive points in the previous column. These two points in turn require the calculation of four consecutive points in the column to the left. Also, it should be noted that calculation of the two consecutive points cannot begin until two sample times after the calculation of the four

consecutive points began in order to allow the first of the four points to reach the end of its shift register.

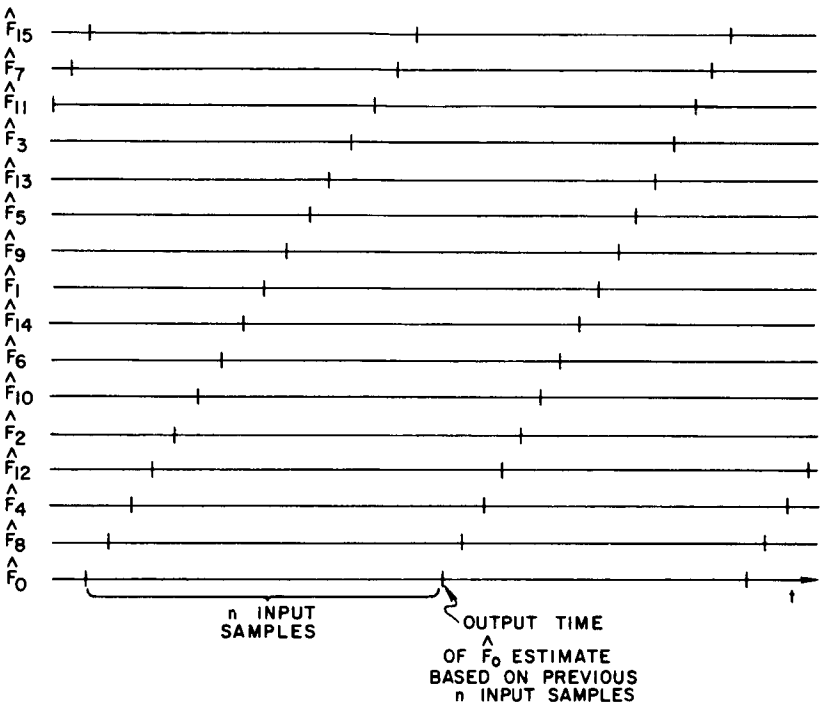


Fig. 31. TIME SEQUENCE OF OUTPUTS AND SAMPLE SETS THEY DEPEND ON.

Similarly, the four points require the calculation of eight consecutive points in the preceding column. Calculation of the four points must be delayed four sample times from the start of the eight consecutive point calculations to allow the four-stage shift register to fill up. Information will be taken from both ends of the four-stage shift register during the next four sample times.

Figure 32 is the flow graph of the real-time algorithm modified so that each output is calculated only once every n input samples. All numbers on this flow graph indicate the sample time at which the operation takes place. All multiplier constants have been left off for the sake of simplicity. They are the same as those in Fig. 25.

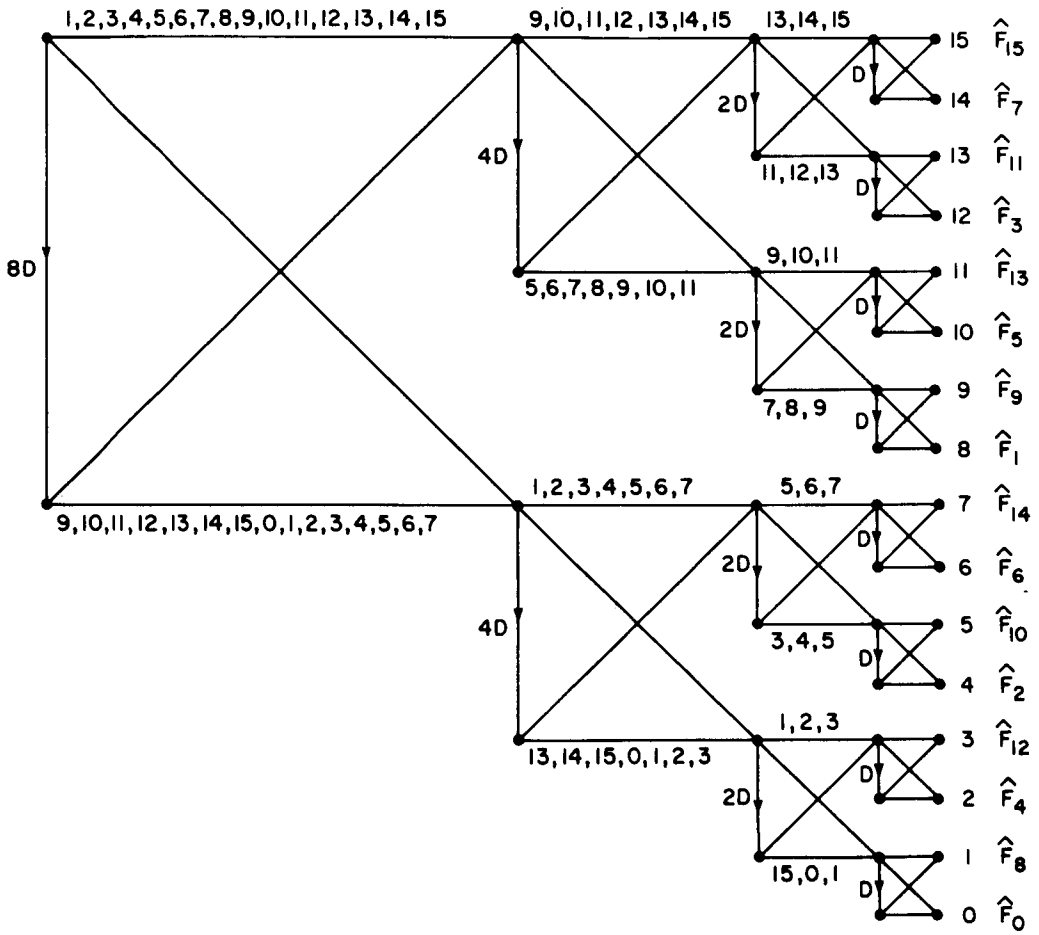


Fig. 32. TIME-SHARED REAL-TIME ALGORITHM (OUTPUTS IN JUMBLED ORDER).

Figure 32 was derived by beginning at the right-hand column with the sample times for the various outputs. We then

work to the left, writing down the sample times at which the inputs to the shift registers must be calculated. For instance, $\hat{F}_2$ and $\hat{F}_{10}$ are calculated at sample times 4 and 5. The input to the shift register from which they are calculated must be calculated at times 4 and 5, and also at time 3 so that the output of the shift register will be meaningful at time 4. Inputs and outputs of the next shift register to the left must be available at times 3, 4, 5, 6, and 7, and since the shift register is of length 2, at times 1 and 2 also. By continuing in this fashion, the sample times at which each of the points of Fig. 32 must be calculated can be obtained.

B. Hardware Reduction

Figure 32 can now be used to minimize the hardware required. Specifically, multiply-add operations which are not performed at the same time can be performed by a single hardware multiply-adder. For instance, a single multiply-adder will suffice for all the operations in the last column.

In all previous columns, two multiply-adders are required per column. In these columns, the top half of all function blocks can be performed by one multiply-adder and the bottom half by a second multiply-adder. This is possible because the different calculations are performed during different sample times. Thus a single hardware multiply-adder may be time-shared among many branches of the flow graph.

Similar flow graphs may be constructed for any number of samples. As long as the outputs are computed in order from

one end of the flow graph to the other, only two multiply-adders will be needed for each column. Thus an analyzer using n data points will require only $2 \log_2 n$ complex multiply-adders.

The multipliers now must be general-purpose multipliers. Since we no longer are always using the same multiplying constant, we cannot simplify the hardware as mentioned in Chapter VI. Also an n-state counter must be employed to keep track of which outputs are being computed and which multiplying constant should be used. Also, the inputs to the multiply-adders must be switched back and forth between the two shift registers of the previous column.

A single binary counter of $\log_2 n$ bits will be sufficient to keep track of all operations. If the states of the counter are decoded in normal binary fashion, the sample time numbers of Fig. 32 will match the state numbers of the binary counter. The frequency-estimate numbers may be obtained by connecting a decode matrix to the outputs of the binary counter with the bit order reversed. Figure 33 illustrates this timing scheme.

C. Output Schemes

If it is desired to produce an oscilloscope output of amplitude vs frequency, digital-to-analog converters may be used as shown in Fig. 34. Since the output of the analyzer is time-shared on a single multiply-adder, a D/A converter may be hardwired to this output and used to control the vertical deflection of the oscilloscope. Another D/A converter is

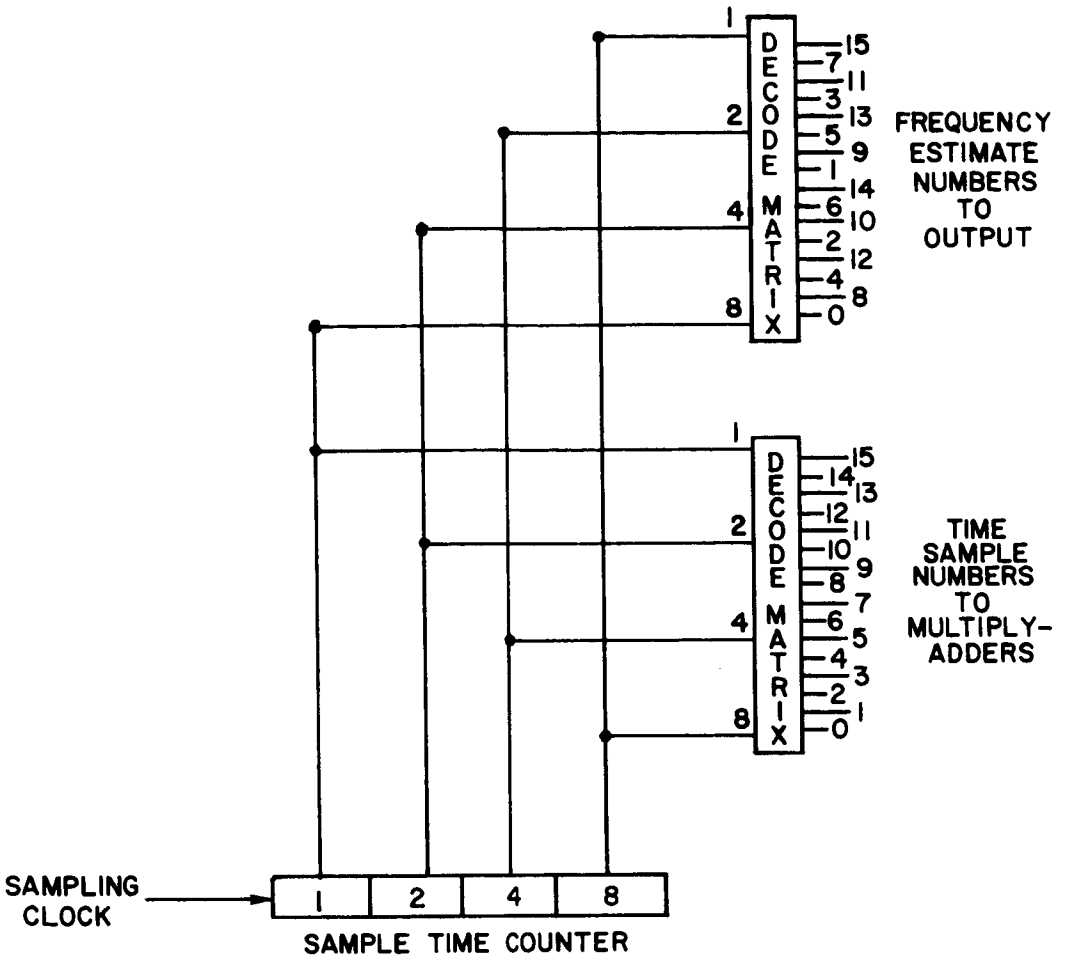


Fig. 33. TIMING CIRCUITRY.

connected to the horizontal deflection input. The resulting horizontal sweep will not be smooth but will jump around in the order of the frequency-estimate outputs.

The oscilloscope provides a convenient way of monitoring the continuously updated outputs of the real-time algorithm. The oscilloscope output also eliminates the need for circuitry to arrange the frequency estimates in a less jumbled order.

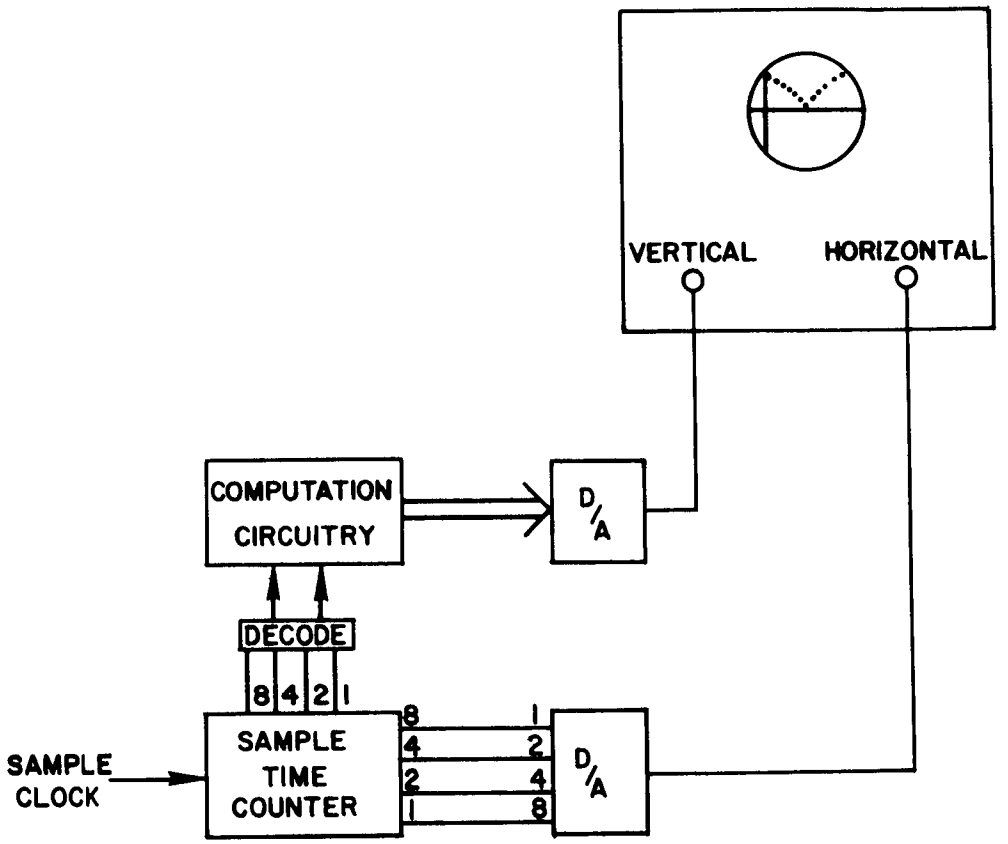


Fig. 34. OSCILLOSCOPE OUTPUT.

Most other useful output schemes will require an n-word register into which the outputs of the analyzer can be addressed using the sample-time counter.

The output of the analyzer may be used directly as the input to a digital computer for further processing in such areas as speech recognition or ECG or EEG classification. In such cases the analyzer output and frequency-estimate number would be taken by the computer in parallel and the reordering would take place as the analyzer output is being stored in the computer memory.

D. Multiplying Constants

In the specialized multipliers of Chapter VI, the multiplying constants were built into our simplification of the multipliers. Now that we are using general-purpose multipliers, the constants must be stored elsewhere and called up as they are required. The easiest way to do this is to use "wired storage." The outputs of the sample-time counter will be decoded and used to furnish appropriate combinations of 1's and 0's to the multiplier inputs. The "appropriate" combinations are determined ahead of time from the flow chart of the calculation. Figure 35 illustrates how a typical wired-storage configuration would look.

Since there will be two complex multipliers in each column, four wired-storage circuits will be needed for each column. This means a total of $4 \log_2 n$ circuits for a complete n-sample analyzer. This additional hardware requirement still is far less than the hardware required to produce all n frequency estimates each sample time.

E. Outputs in Normal Order

Since we are calculating the frequency estimates in rotation, we might inquire as to why we don't calculate them in their normal order. The answer is that we can; however, if we do so, we cannot reduce the hardware requirements as much. This can be seen by referring to the flow graph of the calculation in Fig. 36. This flow chart was derived exactly

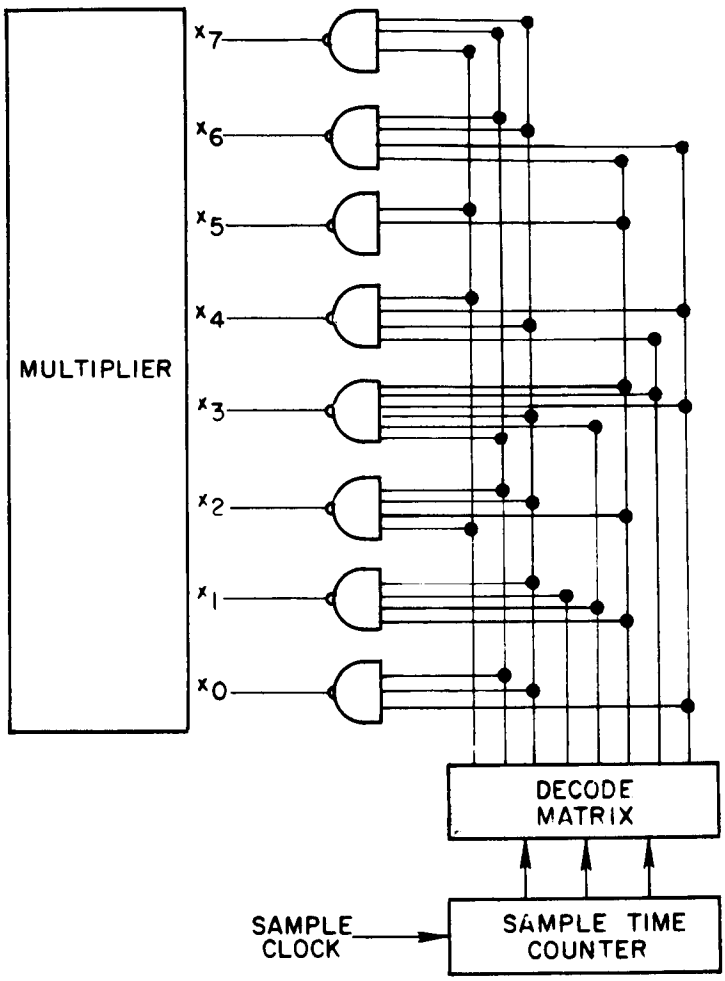


Fig. 35. WIRED STORAGE FOR MULTIPLYING CONSTANTS.

as was the one in Fig. 32 except that the outputs $\hat{F}_0$, $\hat{F}_1$, $\hat{F}_2$,..., $\hat{F}_{15}$ were assigned a different output sequence. Beginning with these assigned sample times and working left, we can determine when all other calculations in the flow chart must be performed. We find that adjacent branches no

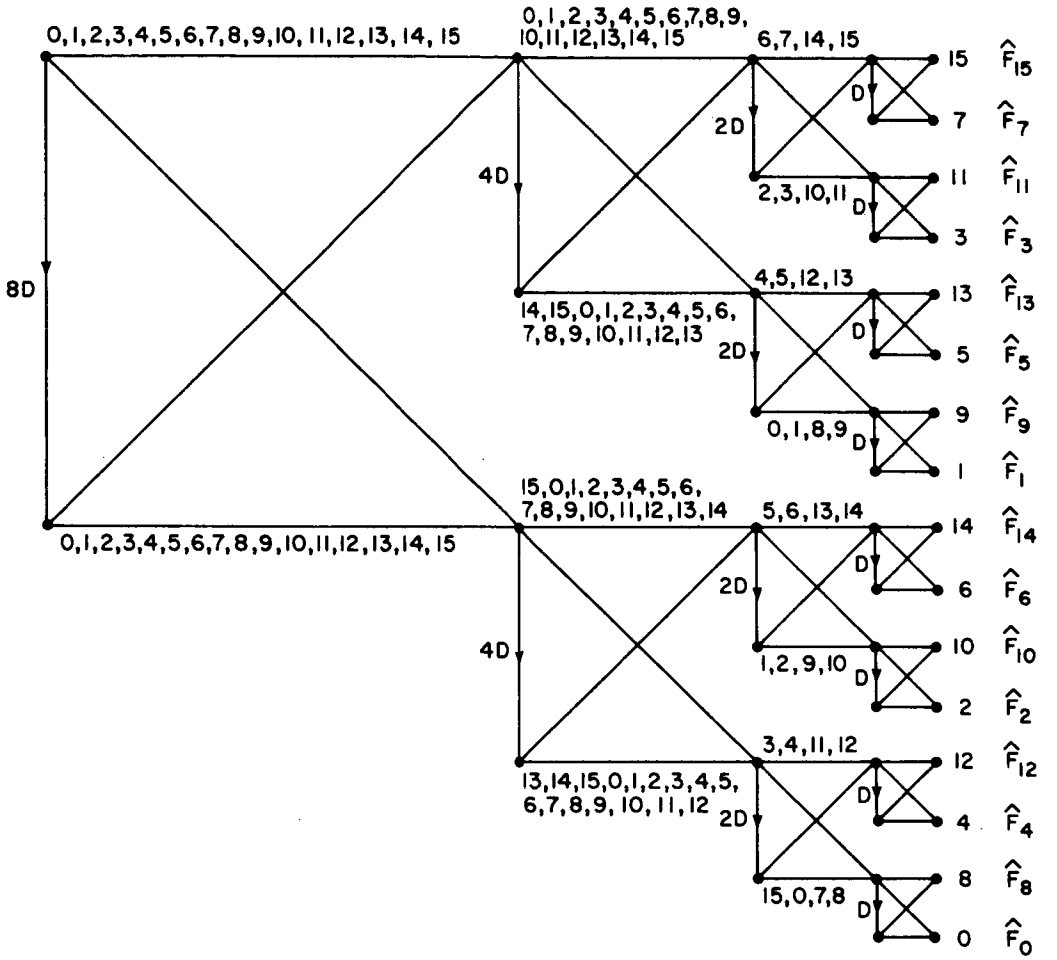


Fig. 36. FLOW CHART FOR FREQUENCY OUTPUTS IN NORMAL ORDER.

longer have sample times in common and thus the number of sample times grows much more rapidly as we progress to the left. The effect of this rapid growth can be easily seen in Fig. 36. As before, the last column requires only one multiply-adder. In the first column back we still need only two multiply-adders because the top four operations may be grouped together, and the bottom four grouped together, with

each group being performed by a single multiply-adder. In the second column back, however, the operations must all be performed continuously and hence four multiply-adders are needed.

When the frequency estimates are output in the normal order, adjacent branches of the computation flow graph do not have any sample times in common. Each operation in the last column will be performed once every n samples. Each operation in the first column back must be performed four times for every n samples. In general, each operation in the r^{th} column back must be performed 2^{2r} times for every n samples.

If we have $n = 2^{10}$ samples, the complete flow chart would have 10 columns. Every operation in the fifth column back $(r = 5)$ would have to be performed $2^{2r} = 2^{10}$ times for every 2^{10} sample times. Thus all operations in the first five columns would have to be performed continuously and the number of multiply-adders in these columns could not be reduced. The last five columns will require $16 + 8 + 4 + 2 + 1 = 31$ multiply-adders, whereas the first five columns will require $32 + 16 + 8 + 4 + 2 = 62$ multiply-adders.

Thus for a sample size of 2^{10} points, outputting the frequency estimates in normal order will require a total of 93 multiply-adders. However, outputting them in the jumbled order of Fig. 32 will require only 20 multiply-adders.

F. Passband Shaping

Time-sharing the multiply-adders to produce only one frequency estimate per sample time has been shown to result in a tremendous savings in hardware. However, passband shaping in the frequency domain is now going to prove more difficult.

There are several reasons for the difficulties. The most obvious is that now we can work with only one estimate at a time. This problem is easily (and expensively) dealt with by furnishing n words of storage for the estimates.

The second problem is that since the estimates occur at different times, their relative phases are not the same as they would be if the estimates occurred at the same time. Equation (10.4) tells us that each estimate may be viewed as a vector rotating at the center frequency of its passband. Figure 37 illustrates such an interpretation of the estimates.

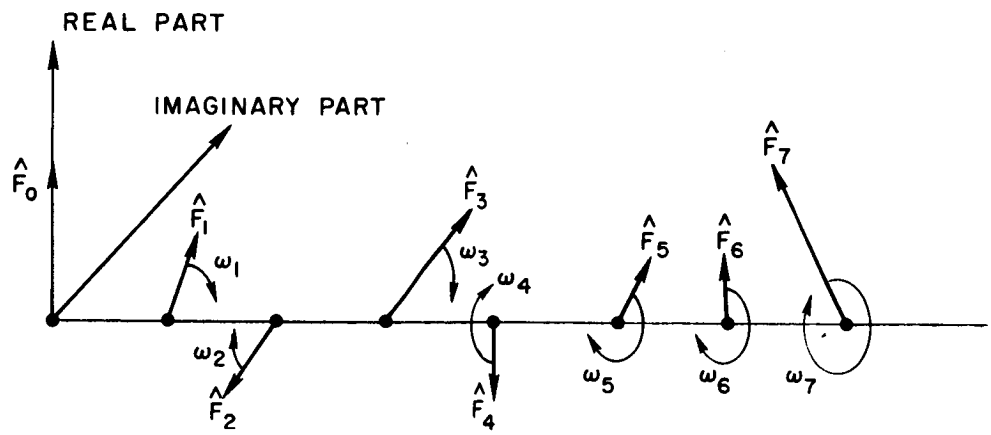


Fig. 37. PHASE VARIATIONS OF FREQUENCY SPECTRUM ESTIMATES.

In order to bring one estimate $\hat{F}(\omega_0,t_1)$ taken at t_1, up to correct phase relationship with another estimate $\hat{F}(\omega_1,t_2)$ taken at t_2, it would be necessary to rotate the first estimate by the correct amount, i.e., to multiply $\hat{F}(\omega_0,t_1)$ by $\exp[-i\omega_0(t_2 - t_1)]$. This multiplication is messy at best and requires a good deal of hardware.

The problem is made worse by the fact that the time spacing between the output of adjacent frequency estimates is not the same for all sets of estimates. If we are to correct the phase of $2n$ estimates to be used in hanning, we will need to do $2n$ additional complex multiplications. This will require two multipliers and considerable indexing and sequencing circuitry since each multiplier will have to choose between n inputs and n multiplying constants.

When the multiplying is all done, we still have only predictions of the estimates. The prediction is based on a sample set which only partially overlaps the set on which the estimate would be calculated. In some cases the overlap is only one or two samples out of n. Thus, if the statistics of the input are time-varying the predictions will be virtually useless.

There is also some uncertainty in the rate of rotation of the estimates since their passbands are of finite width $\Delta\omega$. Thus the uncertainty in the phase of the prediction is of the order of $\pm(\Delta\omega/2)k\tau$, where k is the number of samples ahead that we are predicting. Again using the half-

power bandwidth, we find that our uncertainty in the phase for predicting n samples ahead is one full revolution.

$$\pm \frac{\Delta\omega}{2} n\tau = \pm \frac{2\pi}{2n\tau} n\tau = \pm \pi \qquad (11.1)$$

Thus even for stationary input signals the predictions will be of questionable significance.

G. Example Using Estimate Updating

To illustrate the problems which may arise in trying to update estimates even with signals which are stationary, let us work through the following example. Let us assume we want to update previous estimates of $\hat{F}_2$ and $\hat{F}_4$ in order to hann them with the "present" estimate of $\hat{F}_3$. Referring to Fig. 33, the present is sample time 12. $\hat{F}_2$ and $\hat{F}_4$ will have to be updated by 8 and 10 sample times, respectively.

The hanned estimate would be given by

$$\hat{\hat{F}}_3 = -\frac{1}{2}\hat{F}_2 \exp(-i\omega_2 8\tau) + \hat{F}_3 - \frac{1}{2}\hat{F}_4 \exp(-i\omega_4 10\tau) \qquad (11.2)$$

Let us use a simple cosine of a frequency midway between ω_3 and ω_4 for an input. The cosine can then be expressed as

$$\cos\left(\frac{\omega_3 + \omega_4}{2}\right)t = \frac{\exp\left[i\left(\frac{\omega_3 + \omega_4}{2}\right)t\right] + \exp\left[-i\left(\frac{\omega_3 + \omega_4}{2}\right)t\right]}{2} \qquad (11.3)$$

The true spectrum of the input then will be two real impulses of area 1/2 placed at $\pm(\omega_3 + \omega_4)/2$. This is illustrated in Fig. 38, along with the passband magnitudes of the filters at ω_2, ω_3, and ω_4. The outputs of the filters can be computed using Eq. (10.1). The integrals drop out easily since $F(\omega)$ is an impulse.

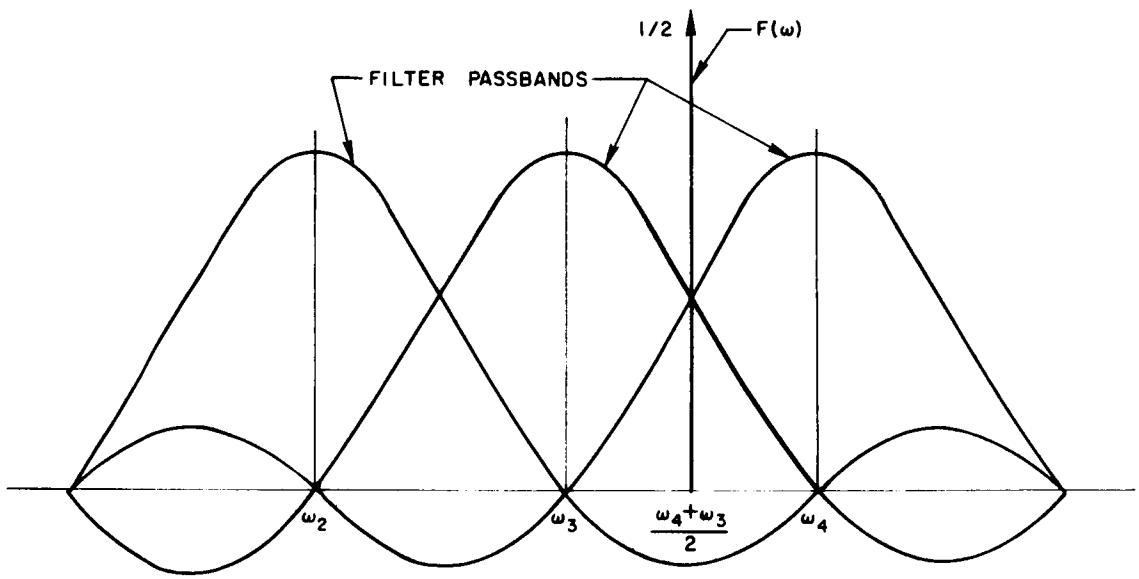


Fig. 38. COSINE OF FREQUENCY $(\omega_3 + \omega_4)/2$ IN THE FREQUENCY DOMAIN.

The output of the filter at ω_3 is given by

$$\hat{F}(\omega_3,\ t = (n-1)\tau + k\tau) = \frac{n}{8\pi^3}\,\frac{1}{2}\,\exp\left[-ik\tau\left(\frac{\omega_3 + \omega_4}{2}\right)\right] \tag{11.4}$$

$$\cdot\ \exp\left[-i\,\frac{n\tau}{2}\left(\omega_3 - \frac{\omega_3 + \omega_4}{2}\right)\right]\ \frac{\sin\frac{n\tau}{2}\left(\omega_3 - \frac{\omega_3 + \omega_4}{2}\right)}{\frac{n\tau}{2}\left(\omega_3 - \frac{\omega_3 + \omega_4}{2}\right)} \tag{11.4}$$

Remembering that the spacing between adjacent frequency estimates is $2\pi/n\tau$, we have

$$\frac{n\tau}{2}\left(\omega_3 - \frac{\omega_3 + \omega_4}{2}\right) = \frac{n\tau}{2}\frac{\pi}{n\tau} = \frac{\pi}{2} \tag{11.5}$$

Substituting (11.5) into (11.4) yields

$$\hat{F}(\omega_3,t) = \frac{n}{8\pi^3}\frac{1}{2}\exp\left[-ik\tau\left(\frac{\omega_3 + \omega_4}{2}\right)\right]\exp\left(-i\frac{\pi}{2}\right)\frac{\sin \pi/2}{\pi/2} \tag{11.6}$$

Simplifying still further gives

$$\hat{F}(\omega_3,t) = -\frac{in}{8\pi^4}\exp\left[-ik\tau\left(\frac{\omega_3 + \omega_4}{2}\right)\right] \tag{11.7}$$

Replacing $k\tau$ by $t - (n-1)\tau$ gives

$$\hat{F}(\omega_3,t) = -\frac{in}{8\pi^4}\exp\left\{-i\left(\frac{\omega_3 + \omega_4}{2}\right)[t - (n-1)\tau]\right\} \tag{11.8}$$

Similarly, the output of the filter at ω_4 is found to be

$$\hat{F}(\omega_4,t) = \frac{in}{8\pi^4}\exp\left\{-i\left(\frac{\omega_3 + \omega_4}{2}\right)[t - (n-1)\tau]\right\} \tag{11.9}$$

For the filter at ω_2 we find

$$\hat{F}(\omega_2,t) = -\frac{in}{24\pi^4}\exp\left\{-i\left(\frac{\omega_3 + \omega_4}{2}\right)[t - (n-1)\tau]\right\} \tag{11.10}$$

Note that all three estimates rotate at the input frequency rather than at their center frequency. If all three estimates

were available concurrently, we could simply hann them as mentioned earlier to yield

$$\hat{\hat{F}}(\omega_3,t) = -\frac{1}{2}\hat{F}(\omega_2,t) + \hat{F}(\omega_3,t) - \frac{1}{2}\hat{F}(\omega_4,t) \qquad (11.11)$$

Using Eqs. (11.8), (11.9), and (11.10), we can write

$$\hat{\hat{F}}(\omega_3,t) = \left(\frac{in}{48\pi^4} - \frac{in}{8\pi^4} - \frac{in}{16\pi^4}\right) \exp\left\{-i\left(\frac{\omega_3+\omega_4}{2}\right)[t-(n-1)\tau]\right\}$$

$$= -\frac{in}{6\pi^4} \exp\left\{-i\left(\frac{\omega_3+\omega_4}{2}\right)[t-(n-1)\tau]\right\} \qquad (11.12)$$

Equation (11.12) gives the correctly hanned output. Unfortunately, when the hardware is time-shared we cannot perform this simple operation since all three estimates are not available at the same time.

The previous estimates for $\hat{F}_2$ and $\hat{F}_4$ may be updated so that they may be hanned with $\hat{F}_3$ according to Eq. (11.2). Equation (11.2) may be expanded by making use of the expressions derived in Eqs. (11.8), (11.9), and (11.10) as follows:

$$\hat{\hat{F}}(\omega_3,t) = -\frac{1}{2}\hat{F}(\omega_2,\ t-8\tau)\exp(-i\omega_2 8\tau)$$

$$+\hat{F}(\omega_3,t) - \frac{1}{2}\hat{F}(\omega_4,\ t-10\tau)\exp(-i\omega_4 10\tau) \qquad (11.13)$$

Expanding $\hat{F}(\omega_2,\ t-8\tau)$ and $\hat{F}(\omega_4,\ t-10\tau)$ yields

$$\hat{F}(\omega_2,\ t-8\tau) = -\frac{in}{24\pi^4}\exp\left\{-i\left(\frac{\omega_3+\omega_4}{2}\right)[t-8\tau-(n-1)\tau]\right\} \qquad (11.14)$$

$$\hat{F}(\omega_4,\ t - 10\tau) = \frac{in}{8\pi^4} \exp\left\{-i\left(\frac{\omega_3 + \omega_4}{2}\right)[t - 10\tau - (n-1)\tau]\right\} \tag{11.15}$$

Putting (11.14) and (11.15) back into (11.13) gives

$$\begin{aligned}\hat{\hat{F}}(\omega_3,t) = {} & \frac{1}{2}\frac{in}{24\pi^4} \exp\left\{-i\left(\frac{\omega_3 + \omega_4}{2}\right)[t - (n-1)\tau]\right\} \\ & \cdot \exp\left[-i8\tau\left(\omega_2 - \frac{\omega_3 + \omega_4}{2}\right)\right] \\ & - \frac{in}{8\pi^4} \exp\left\{-i\left(\frac{\omega_3 + \omega_4}{2}\right)[t - (n-1)\tau]\right\} \\ & - \frac{1}{2}\frac{in}{8\pi^4} \exp\left\{-i\left(\frac{\omega_3 + \omega_4}{2}\right)[t - (n-1)\tau]\right\} \\ & \cdot \exp\left[-i\ 10\tau\left(\omega_4 - \frac{\omega_3 + \omega_4}{2}\right)\right]\end{aligned} \tag{11.16}$$

Collecting like terms yields

$$\begin{aligned}\hat{\hat{F}}(\omega_3,t) = {} & \left\{\frac{in}{48\pi^4} \exp\left[-i\ 8\tau\left(\omega_2 - \frac{\omega_3 + \omega_4}{2}\right)\right] - \frac{in}{8\pi^4}\right. \\ & \left. - \frac{in}{16\pi^4} \exp\left[-i\ 10\tau\left(\omega_4 - \frac{\omega_3 + \omega_4}{2}\right)\right]\right\} \\ & \cdot \exp\left\{-i\left(\frac{\omega_3 + \omega_4}{2}\right)[t - (n-1)\tau]\right\}\end{aligned} \tag{11.17}$$

Equation (11.17) differs from the correctly hanned estimate of Eq. (11.12). The difference is due to the fact that the estimates were rotated at their center frequency to update them rather than at the correct frequency $(\omega_3 + \omega_4)/2$. The

result is an error in the relative phases of the components of the hanned estimate. In this particular example the phase angle errors are

$$-8\tau\left(\omega_2 - \frac{\omega_3 + \omega_4}{2}\right) = \frac{3}{2}\pi \qquad (11.18a)$$

$$-10\tau\left(\omega_4 - \frac{\omega_3 + \omega_4}{2}\right) = -\frac{5}{8}\pi \qquad (11.18b)$$

The effect of these relative phase errors is shown graphically in Fig. 39. Both hanned estimates--the one correctly done and the one done by updating--rotate at the frequency of the input cosine wave. However, their magnitudes are drastically different as shown in Fig. 39.

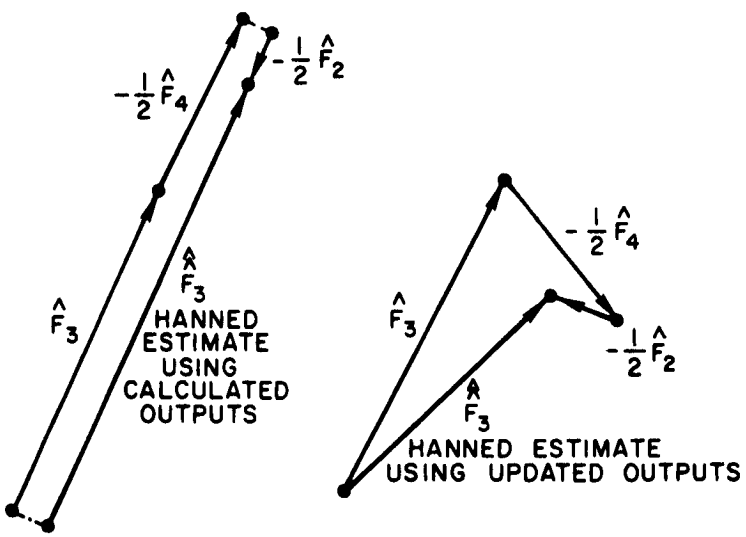


Fig. 39. EXAMPLE OF RELATIVE PHASE ERRORS IN HANNED ESTIMATES USING UPDATING.

This example demonstrates quite nicely that, even if the input is stationary, trying to update previous frequency estimates by rotating them at their center frequency may lead to gross errors. It is for this reason that the author feels that if passband shaping is to be done in the frequency domain, it must be done using estimates which are calculated at the same time, not estimates which have been rotated to update them.

H. Algorithm Modification To Permit Hanning

As a consequence of the above problems involved with updating old estimates, we are led to considering calculating several estimates each sample time rather than one. Figure 40 was constructed by assigning the same output times for the frequency estimates as was found most efficient in Fig. 32. Then the diagram was modified such that, for every frequency estimate, the two adjacent frequency estimates were calculated at the same time. This was done so that three adjacent outputs would be available concurrently to permit hanning. Working backward as before, sample times were determined for all other operations.

The result of starting with three sample times instead of one is that there is considerably more duplication of sample times in the various branches of the flow graph. Consequently, fewer branches may be grouped together and performed by a single hardware multiply-adder.

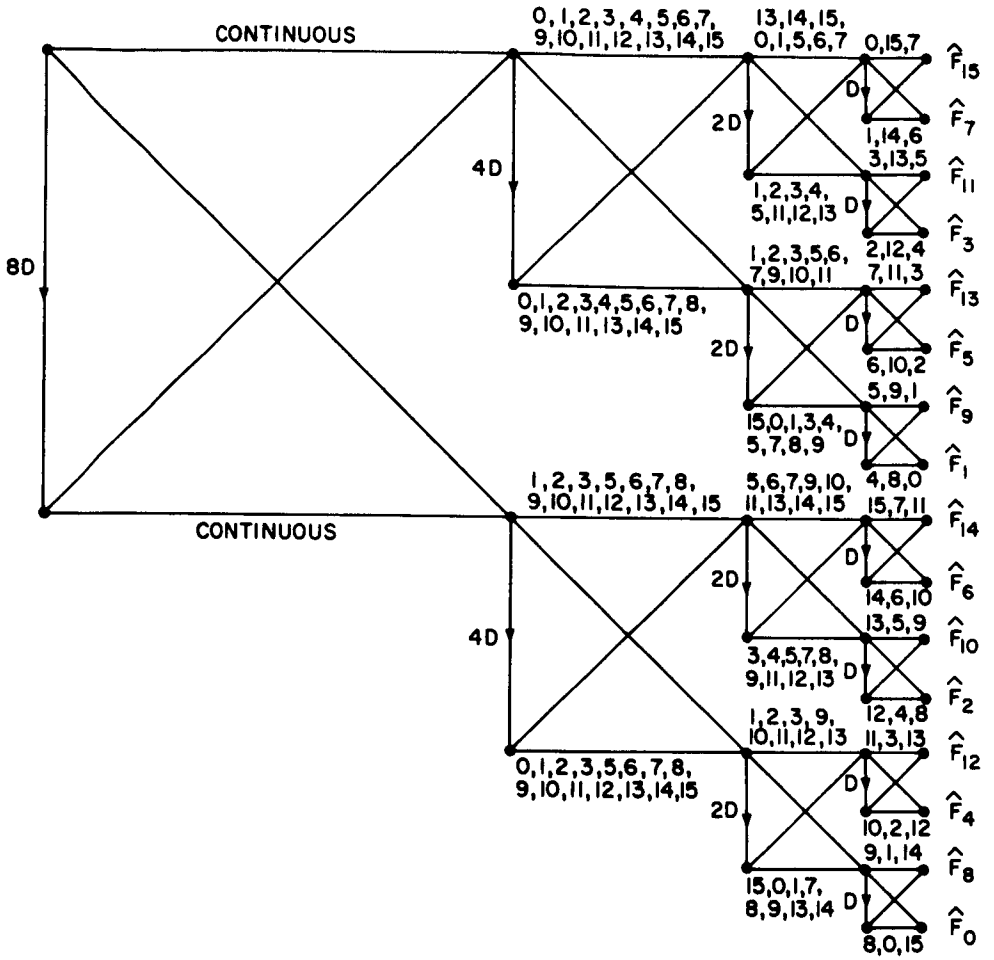


Fig. 40. COMPUTATION FLOW GRAPH TO BE USED WITH HANNING (OUTPUTS IN JUMBLED ORDER).

It was decided to try another output order in hopes of achieving more efficient operation. For this purpose Fig. 41 was constructed assuming a normal order for the frequency estimates; that is, $\hat{F}_0$ is desired at sample time 0; $\hat{F}_1$ at time 1; $\hat{F}_2$ at 2; etc. Each estimate has been calculated three consecutive times so that it may be used in hanning with the estimate before it and the estimate after it.

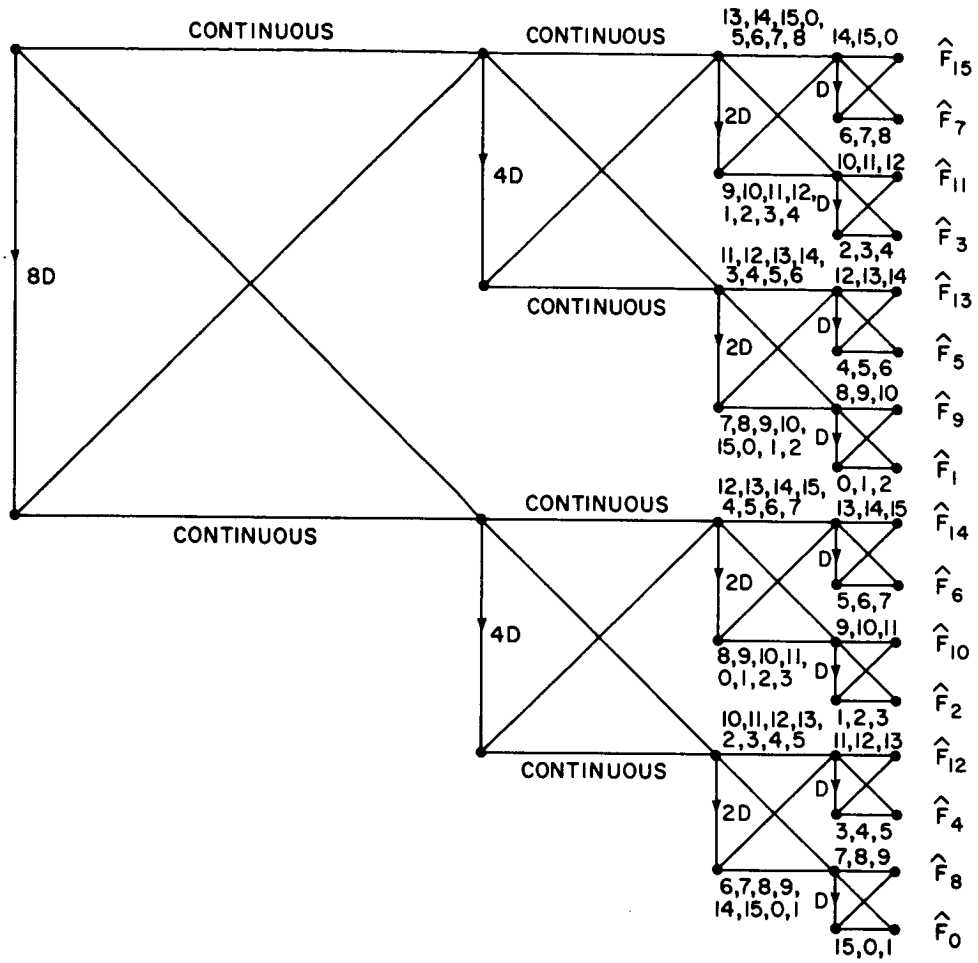


Fig. 41. COMPUTATION FLOW GRAPH TO BE USED WITH HANNING (OUTPUTS IN NORMAL ORDER).

A greater savings in hardware is realized from Fig. 41 than from Fig. 40. In the latter figure the three estimate calculation times were not consecutive; consequently in the first column to the left, three additional calculation times were needed in order to load the shift registers. In Fig. 41 only two additional calculation times are needed.

In analyzing Fig. 41 we note that each operation in the last column is performed in a string of three consecutive times. Sample times for operations in the same box are $n/2$ apart. In the first column back each operation is performed in two strings of four consecutive sample times. The spacing between strings is $n/2$ sample times. Other strings in the same box are $n/4$ away.

In general, in the r^{th} row back, operations will be performed in strings of length as given in Eq. (11.19).

$$\text{String length} = L = 3 + 1 + 2 + 4 + \ldots + 2^{r-1}$$
$$= L = 3 + 2^r - 1 = 2 + 2^r \qquad (11.19)$$

The number of strings N is given by

$$N = 2^r \qquad (11.20)$$

The strings will be spaced $n/2^r$ apart. The other operation in the same box will have similar strings spaced $n/2^r$ apart and offset from these strings by $n/2^{r+1}$. There will be no overlapping of sample times within the box as long as

$$2 + 2^r \leq \frac{n}{2^{r+1}} \qquad (11.21)$$

Since $n = 2^m$ we have

$$2^{r+2} + 2^{2r+1} \leq 2^m \qquad (11.22)$$

We can make use of the fact that m is an integer and write

$$2^{2r+1} < 2^m \tag{11.23}$$

Stating it another way, for an m-column flow graph we may use a single multiply-adder for the operations on the top and bottom of each box in the $(m/2)+1^{st}$ column. Furthermore, all operations which are generated from each of these boxes will require only one multiply-adder per column. For Fig. 41, then, the top and bottom of each box in column 3 may be performed by time-sharing a single multiply-adder. Also, the two boxes in column 4, which are generated from the outputs of the top box in column 3, may be implemented by time-sharing a single hardware multiply-adder.

The $(m/2)^{th}$ column will contain $2^{m/2}$ boxes, or 2^m multiply-adders. Using the single multiplier per box realization of Fig. 20, this number may be cut to $2^{m/2}$ multipliers and 2^m adders. Thus the total number M of multipliers can be held to

$$\begin{aligned} M &= 1 + 2 + 4 + \ldots + 2^{m/2} + \frac{m}{2} 2^{m/2} \\ &= \left(\frac{m}{2} + 2\right) 2^{m/2} - 1 \end{aligned} \tag{11.24}$$

Equation (11.24) then gives an upper limit on the number of multipliers that will be needed to produce an output suitable for hanning, given any number $n = 2^m$ of samples.

The operation grouping mentioned above is still not as efficient as it can be, as is evident from Fig. 41. According

to Eq. (11.24) we can hold the number of multiply-adders in column 4 to $2^{m/2} = 2^2 = 4$. However, if we look at the sample-time numbers in this column, we see that no number appears more than three times. This means that we need never perform more than three multiply-adds simultaneously, and hence we need only three hardware multiply-adders to implement this last column.

In order to get most efficient operation grouping we must group so that each multiply-adder is always working each sample time. For instance, the multiply-adder which is used to calculate $\hat{F}_{15}$ during sample times 14, 15, and 0 should be started off on the calculation of $\hat{F}_2$ during sample times 1, 2, 3. During sample times 4, 5, 6 this multiply-adder will be used to calculate $\hat{F}_5$. Continuing in this fashion we find that we do not return to our starting point until $3n$ sample times later when this multiply-adder has been used to calculate all n of the frequency estimates. In the last column we had operations to be performed in strings of three consecutive sample times. Since we had 16 such strings, it took $3 \times 16 = 48$ sample times to return to the original starting point.

In the first column back, there are still 16 strings but now they are 4 sample times long. In the second column back, there will still be 16 strings but they will be 6 sample times long to allow time to fill up the shift registers.

In the right-hand columns, the minimum number of multiply-

adders necessary will be equal to the number of sample times in the strings. This is true since the number of strings remains constant at n and the number of total operations in a column in n sample times is equal to the number of strings multiplied by the number of sample times in the string. In the left-hand columns the minimum number of multiply-adders will be

$$M = 2^r \tag{11.25}$$

where r is the column number.

In the right-hand columns we have

$$\begin{aligned} M &= 3 + (1 + 2 + 4 + \ldots + 2^{m-r-1}) \\ &= 3 + (2^{m-r} - 1) \\ &= 2 + 2^{m-r} \end{aligned} \tag{11.26}$$

The crossover point occurs where $r = m/2$. Hence, the first columns will run continuously with a total number of multiply-adders given by

$$2 + 4 + \ldots + 2^{m/2} = 2(2^{m/2} - 1) \tag{11.27}$$

The last $m/2$ columns will run time-shared with a total number of multiply-adders given by

$$2^{\frac{m}{2}} + \sum_{r=\frac{m}{2}+1}^{m} 2^{m-r} = m + 2^{m/2} - 1 \tag{11.28}$$

Thus the minimum total number of multiply-adders needed to produce three simultaneous frequency estimates suitable for hanning is

$$\begin{aligned} & 2(2^{m/2} - 1) + m + 2^{m/2} - 1 \\ & = 3 \cdot 2^{m/2} + m - 3 \\ & = 3(2^{m/2} - 1) + m \end{aligned} \qquad (11.29)$$

Thus, if hanning is desired it is seen that the number of multiply-adders grows much more rapidly than the $2m$ rate necessary for unhanned signals.

Chapter XII

SUMMARY AND SUGGESTIONS

A. Summary

This research has developed a new algorithm for spectral analysis in real time. This new algorithm is intended for implementation by special-purpose circuitry using currently available integrated circuits. Maximum sampling rates above 1 MHz can be realized in continuous real-time operation. This is much faster than any other existing algorithm can operate.

A closed-form analytical expression has been developed for the passband characteristics of the discrete Fourier transform operation. This expression was used to evaluate in detail the effectiveness of several time-domain and frequency-domain operations aimed at improving the passband characteristics.

A hardware design has been presented which makes use of certain novel features of this new real-time algorithm. Among these features is the use of shift registers in which only the inputs and outputs are available. This permits use of very long integrated circuit shift registers in very small packages since very few inputs and outputs are required. This feature would not be an advantage in realizing this algorithm on a general-purpose computer, but it definitely is for realizing it in special-purpose hardware.

B. Suggestions for Further Research

A number of areas exist where it appears that further research would be profitable. For example, it should be possible to combine two consecutive frequency estimates at any one frequency to obtain another estimate for this frequency which would be based on $2n$ rather than n input samples. Consequently this new frequency estimate should have a passband which is only half as wide as those of the original estimates.

Another possibility is the use of the spectrum analyzer as a digital frequency synthesizer. Notice that if we put an impulse into the analyzer, $\hat{F}_0$ will output n samples of dc. $\hat{F}_1$ will output n samples (one complete cycle) of a complex exponential (sines and cosines). $\hat{F}_2$ will output two cycles; $\hat{F}_3$, three cycles; and so forth. After n samples all outputs return to zero. If impulses are input once every n sample times, the output will be continuous sine waves whose periods are submultiples of $n\tau$.

The time-shared algorithm computes only one frequency estimate every sample time. Hence operations in the frequency domain for improving the passband characteristics are difficult. The author feels that hardware computation of several frequency estimates concurrently is economically unattractive. If the frequency estimates are to be operated on further in a computer after the fact, it may be reasonable to take two consecutive estimates for any given frequency and use these

to interpolate somehow to get estimates at all sample times in between. Then with this complete set of estimates, pass-band shaping could be done after the fact. Interpolation is discussed further in Appendix B.

Appendix A

CONVOLUTION THEOREM

The Fourier transform and its inverse as used in this study are given by

$$F(\omega) = \int_{-\infty}^{\infty} f(t)\, e^{-i\omega t}\, dt \qquad (A.1)$$

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega)\, e^{i\omega t}\, d\omega \qquad (A.2)$$

The transform of a product $f_1(t)\, f_2(t)$ is then

$$F_1F_2(\omega) = \int_{-\infty}^{\infty} f_1(t)\, f_2(t)\, e^{-i\omega t}\, dt \qquad (A.3)$$

Let us express $f_1(t)$ and $f_2(t)$ in terms of their inverse transforms.

$$F_1F_2(\omega) = \int_{-\infty}^{\infty} \left[\frac{1}{2\pi} \int_{-\infty}^{\infty} F_1(\omega_1)\, e^{i\omega_1 t}\, d\omega_1\right] \cdot \left[\frac{1}{2\pi} \int_{-\infty}^{\infty} F_2(\omega_2)\, e^{i\omega_2 t}\, d\omega_2\right] e^{-i\omega t}\, dt \qquad (A.4)$$

Rearranging the order of integration yields

$$F_1F_2(\omega) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} F_1(\omega_1) \int_{-\infty}^{\infty} F_2(\omega_2) \int_{-\infty}^{\infty} \exp[i(\omega_1+\omega_2-\omega)t]\, dt\, d\omega_2\, d\omega_1 \qquad (A.5)$$

The inside integral over t is identically zero except where $\omega_1 + \omega_2 - \omega = 0$, where it gives

$$\int_{-\infty}^{\infty} \exp[i(\omega_1+\omega_2-\omega)t]\, dt = 2\pi\delta(\omega_1+\omega_2-\omega) \qquad (A.6)$$

Then

$$F_1F_2(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F_1(\omega_1) \int_{-\infty}^{\infty} F_2(\omega_2)\ \delta(\omega_1+\omega_2-\omega)\ d\omega_2\ d\omega_1 \qquad (A.7)$$

$$F_1F_2(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F_1(\omega_1)\ F_2(\omega-\omega_1)\ d\omega_1 \qquad (A.8)$$

Equation (A.8) expresses the convolution theorem that the transform of a product is the convolution of the individual spectra.

The convolution of two functions is often written in operational notation as

$$f_1(x) \otimes f_2(x) = \int_{-\infty}^{\infty} f_1(x-y)\ f_2(y)\ dy \qquad (A.9)$$

Using the notation of (A.9) and using a double-headed arrow to indicate the taking of a transform we may write

$$f_1(t) \longleftrightarrow F_1(\omega)$$

$$f_2(t) \longleftrightarrow F_2(\omega)$$

$$f_1(t)\ f_2(t) \longleftrightarrow \frac{1}{2\pi} F_1(\omega) \otimes F_2(\omega) \qquad (A.10)$$

In a similar manner we can obtain

$$f_1(t) \otimes f_2(t) \longleftrightarrow F_1(\omega)\, F_2(\omega) \qquad (A.11')$$

Appendix B

INTERPOLATION

The time-shared real-time algorithm produces each frequency estimate only once for every n input samples. We have argued that, by looking at only every n^{th} output for a given frequency, we can obtain all the information that is present at that frequency. Or to put it another way--if we have two consecutive estimates (which are n sample times apart), we can interpolate in some fashion to obtain estimates for all sample times in between.

From Eq. (10.1) we can express the time-varying output of any frequency estimate as

$$\hat{F}(\omega_0,t) = \frac{n}{8\pi^3} \int_{-\infty}^{\infty} F(\omega) \exp\left\{-i\omega[t - (n-1)\tau]\right\} \cdot \exp\left[-i \frac{n\tau}{2} (\omega_0 - \omega)\right] \frac{\sin \frac{n\tau}{2} (\omega_0 - \omega)}{\frac{n\tau}{2} (\omega_0 - \omega)} d\omega \tag{B.1}$$

At a time $n\tau$ seconds later the estimate will be

$$\hat{F}(\omega_0,\ t + n\tau) = \frac{n}{8\pi^3} \int_{-\infty}^{\infty} F(\omega) \exp\left\{-i\omega[t + n\tau - (n-1)\tau]\right\} \cdot \exp\left[-i \frac{n\tau}{2} (\omega_0 - \omega)\right] \frac{\sin \frac{n\tau}{2} (\omega_0 - \omega)}{\frac{n\tau}{2} (\omega_0 - \omega)} d\omega \tag{B.2}$$

If we assume that the input is a sinusoid that falls within the 3 dB passband of the filter, then both the input and output will be of a frequency $\omega_0 \pm \pi/n\tau$. The input will be a simple real sinusoid of this frequency, and the output will be a complex exponential which rotates at this frequency.

Between t and $t + n\tau$ the output will rotate $\omega_0 n\tau \pm \pi$ radians. Recalling that $\omega_0 = j2\pi/n\tau$, we see that the output rotates j complete revolutions $\pm$ 1/2 revolution. The rotation of the output is illustrated in Fig. 42.

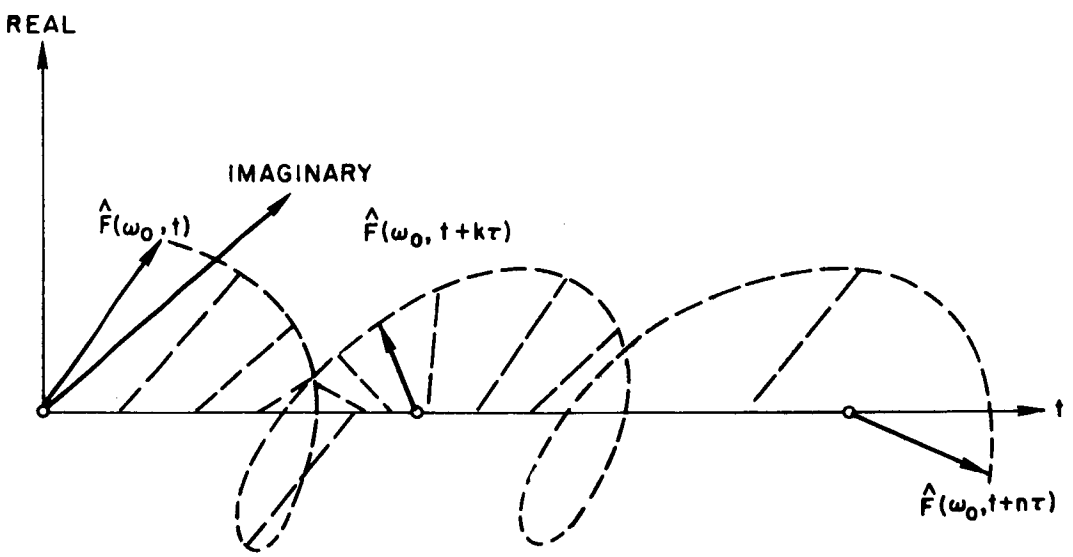


Fig. 42. ROTATION OF FREQUENCY ESTIMATE OUTPUT.

We can use the computed outputs at times t and $t + n\tau$ to interpolate and obtain an output estimate for some time $t + k\tau$ in between. We will simply interpolate linearly both

the magnitude and angle of the output. Doing so yields for the magnitude

$$|\hat{F}(\omega_0,\ t + k\tau)| = |\hat{F}(\omega_0,t)| + \frac{k\tau}{n\tau}\left[|\hat{F}(\omega_0,\ t + n\tau)| - |\hat{F}(\omega_0,t)|\right]$$

$$= \frac{n - k}{n}\,|\hat{F}(\omega_0,t)| + \frac{k}{n}\,|\hat{F}(\omega_0,\ t + n\tau)| \quad \text{(B.3)}$$

The angle of the interpolated output is given by

$$\angle\hat{F}(\omega_0,\ t + k\tau) = \angle\hat{F}(\omega_0,t) + \frac{k\tau}{n\tau}\left[\omega_0 n\tau + \angle\hat{F}(\omega_0,\ t + n\tau) - \angle\hat{F}(\omega_0,t)\right]$$

$$= \frac{n - k}{n}\angle\hat{F}(\omega_0,t) + \frac{k}{n}\angle\hat{F}(\omega_0,\ t + n\tau) + \omega_0 k\tau \quad \text{(B.4)}$$

It should be remembered that the angle of Eq. (B.4) was arrived at by assuming that the input sinusoid fell within the 3 dB passband of the ω_0 filter. If it does not, then the angle of the interpolated frequency estimate will be incorrect. Since the interpolated output must pass through the points $\hat{F}(\omega_0,t)$ and $\hat{F}(\omega_0,\ t + n\tau)$, the angle at time $t + n\tau$ can vary only by an integral number of revolutions. Consequently, at time $t + k\tau$ the possible errors in angle are

$$\pm 2\pi\,\frac{k}{n},\ \pm 4\pi\,\frac{k}{n},\ \pm 6\pi\,\frac{k}{n},\ \ldots$$

Let us use Eq. (10.1) to express the interpolated output in terms of the true spectrum.

$$|\hat{F}(\omega_0,\ t + k\tau)| = \frac{n-k}{n}\frac{n}{8\pi^3}\left|\int_{-\infty}^{\infty} F(\omega)\ \exp[-i\,\tfrac{n\tau}{2}(\omega_0 - \omega)]\ \frac{\sin\frac{n\tau}{2}(\omega_0 - \omega)}{\frac{n\tau}{2}(\omega_0 - \omega)}\,d\omega\right|$$

$$+ \frac{k}{n}\frac{n}{8\pi^3}\left|\int_{-\infty}^{\infty} F(\omega)\ \exp(-in\tau\omega)\ \exp[-i\,\tfrac{n\tau}{2}(\omega_0 - \omega)]\ \frac{\sin\frac{n\tau}{2}(\omega_0 - \omega)}{\frac{n\tau}{2}(\omega_0 - \omega)}\,d\omega\right|$$

$$= \frac{n-k}{8\pi^3}\left|\int_{-\infty}^{\infty} F(\omega)\ \exp[-i\,\tfrac{n\tau}{2}(\omega_0 - \omega)]\ \frac{\sin\frac{n\tau}{2}(\omega_0 - \omega)}{\frac{n\tau}{2}(\omega_0 - \omega)}\,d\omega\right|$$

$$+ \frac{k}{8\pi^3}\left|\int_{-\infty}^{\infty} F(\omega)\ \exp[-i\,\tfrac{n\tau}{2}(\omega_0 - \omega)]\ \frac{\sin\frac{n\tau}{2}(\omega_0 - \omega)}{\frac{n\tau}{2}(\omega_0 - \omega)}\,d\omega\right| \qquad \text{(B.5)}$$

It should be noted at this point that the interpolated magnitude of the output as given by Eq. (B.5) is obtained by a nonlinear operation on the input. This means that if we put in a signal which is the sum of two sinusoids, the output will <u>not</u> be simply the sum of the individual responses to the sinusoids. Consequently we cannot define a passband characteristic for the interpolated output as we did for the computed outputs. We can, however, use a single cosine wave to test and partially define the frequency response of the interpolated output. For a cosine wave of frequency ω_1 we have

$$F(\omega) = \pi\delta(\omega - \omega_1) + \pi\delta(\omega + \omega_1) \qquad \text{(B.6)}$$

Inserting (B.6) into (B.5) yields

$$|\hat{F}(\omega_0,\ t + k\tau)| = \frac{n - k}{8\pi^2} \left| \exp[-i \tfrac{n\tau}{2} (\omega_0 - \omega_1)] \frac{\sin \frac{n\tau}{2} (\omega_0 - \omega_1)}{\frac{n\tau}{2} (\omega_0 - \omega_1)} \right.$$

$$\left. + \exp[-i \tfrac{n\tau}{2} (\omega_0 + \omega_1)] \frac{\sin \frac{n\tau}{2} (\omega_0 + \omega_1)}{\frac{n\tau}{2} (\omega_0 + \omega_1)} \right|$$

$$+ \frac{k}{8\pi^2} \left| \exp[-i \tfrac{n\tau}{2} (\omega_0 + \omega_1)] \frac{\sin \frac{n\tau}{2} (\omega_0 - \omega_1)}{\frac{n\tau}{2} (\omega_0 - \omega_1)} \right.$$

$$\left. + \exp[-i \tfrac{n\tau}{2} (\omega_0 - \omega_1)] \frac{\sin \frac{n\tau}{2} (\omega_0 + \omega_1)}{\frac{n\tau}{2} (\omega_0 + \omega_1)} \right| \quad \text{(B.7)}$$

In the vicinity of ω_0 we have

$$\left| \exp[-i \tfrac{n\tau}{2} (\omega_0 - \omega_1)] \frac{\sin \frac{n\tau}{2} (\omega_0 + \omega_1)}{\frac{n\tau}{2} (\omega_0 + \omega_1)} \right| << \left| \exp[-i \tfrac{n\tau}{2} (\omega_0 + \omega_1)] \frac{\sin \frac{n\tau}{2} (\omega_0 - \omega_1)}{\frac{n\tau}{2} (\omega_0 - \omega_1)} \right| \quad \text{(B.8a)}$$

$$\left| \exp[-i \tfrac{n\tau}{2} (\omega_0 + \omega_1)] \frac{\sin \frac{n\tau}{2} (\omega_0 + \omega_1)}{\frac{n\tau}{2} (\omega_0 + \omega_1)} \right| << \left| \exp[-i \tfrac{n\tau}{2} (\omega_0 - \omega_1)] \frac{\sin \frac{n\tau}{2} (\omega_0 - \omega_1)}{\frac{n\tau}{2} (\omega_0 - \omega_1)} \right| \quad \text{(B.8b)}$$

Consequently, near ω_0 we can write

$$|\hat{F}(\omega_0,\ t + k\tau)| \approx \frac{n}{8\pi^2} \left| \frac{\sin \frac{n\tau}{2} (\omega_0 - \omega_1)}{\frac{n\tau}{2} (\omega_0 - \omega_1)} \right| \quad \text{(B.9)}$$

Similarly, near $-\omega_0$ we have

$$|\hat{F}(\omega_0,\ t + k\tau)| \approx \frac{n}{8\pi^2} \left| \frac{\sin \frac{n\tau}{2} (\omega_0 + \omega_1)}{\frac{n\tau}{2} (\omega_0 + \omega_1)} \right| \quad \text{(B.10)}$$

Figure 43 illustrates this magnitude "passband characteristic."

It should be remembered, as mentioned above, that the interpolated output is the result of a nonlinear operation. Consequently a true passband characteristic cannot be defined for it. Figure 43, then, should be thought of only as a first-order approximation to a passband characteristic.

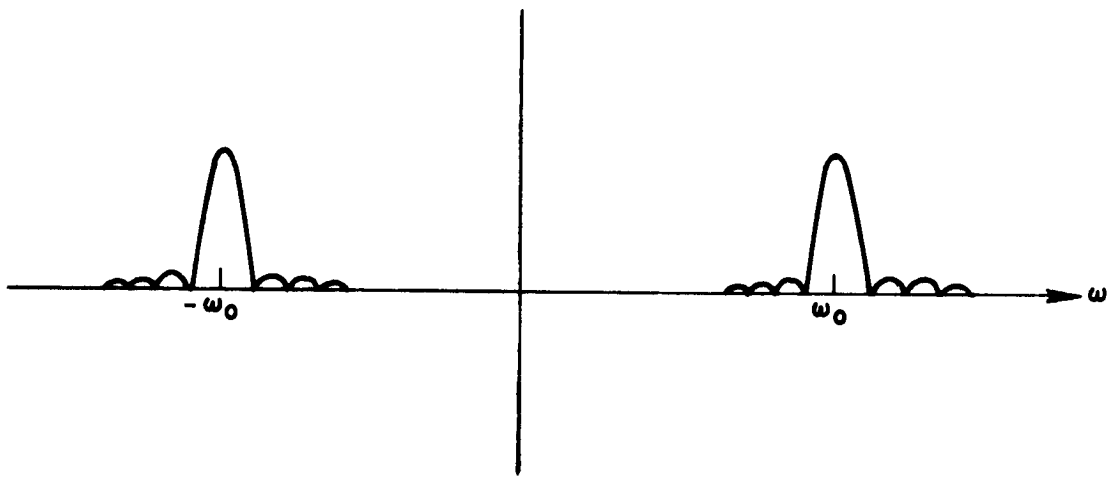


Fig. 43. MAGNITUDE "PASSBAND CHARACTERISTIC" OF INTERPOLATED OUTPUT.

BIBLIOGRAPHY

Allen, J. L., "The Theory of Array Antennas," Technical Report No. 323, M.I.T. Lincoln Laboratory, Cambridge, Mass., 25 Jul 1963.

Aseltine, J. A., Transform Method in Linear System Analysis, McGraw-Hill Book Company, New York, 1958.

Bergland, G. D., and H. W. Hale, "Digital Real-Time Spectral Analysis," IEEE Trans. Electronic Computers, EC-16, 2, Apr 1967.

Bingham, C., M. D. Godfrey, and J. W. Tukey, "Modern Techniques of Power Spectrum Estimation," IEEE Trans. Audio and Electroacoustics, AU-15, 2, Jun 1967.

Blackman, R. B., and J. W. Tukey, The Measurement of Power Spectra, Dover Book Company, New York, 1959.

Bogert, B. P., and E. Parzen, "Informal Comments on the Uses of Power Spectrum Analysis," IEEE Trans. Audio and Electroacoustics, AU-15, 2, Jun 1967.

Bracewell, R. M., The Fourier Transform and Its Applications, McGraw-Hill Book Company, New York, 1965.

Chu, T., Digital Computer Design Fundamentals, McGraw-Hill Book Company, New York, 1962.

Cooley, J. W., P. A. W. Lewis, and P. D. Welch, "Historical Notes on the Fast Fourier Transform," IEEE Trans. Audio and Electroacoustics, AU-15, 2, Jun 1967a.

Cooley, J. W., P. A. W. Lewis, and P. D. Welch, "Application of the Fast Fourier Transform to Computation of Fourier Integrals, Fourier Series, and Convolution Integrals," IEEE Trans. Audio and Electroacoustics, AU-15, 2, Jun 1967b.

Cooley, J. W., and J. W. Tukey, "An Algorithm for the Machine Calculation of Complex Fourier Series," Math. Computation, 19, 90, Apr 1965.

Daley, F. D., Jr., "Analog-to-Digital Conversion Techniques," Electro-Technology, May 1967.

Flanagan, J. L., "Spectrum Analysis in Speech Coding," IEEE Trans. Audio and Electroacoustics, AU-15, 2, Jun 1967.

G-AE Subcommittee on Measurement Concepts, "What Is the Fast Fourier Transform?" IEEE Trans. Audio and Electroacoustics, AU-15, 2, Jun 1967.

Gentleman, W. M., and G. Sande, "Fast Fourier Transforms--for Fun and Profit," AFIPS Fall Joint Computer Conf. Proc., 29, Spartan Books, Washington, D.C., 1966, pp. 563-578.

Helms, H. D., "Fast Fourier Transform Method of Computing Difference Equations and Simulating Filters," IEEE Trans. Audio and Electroacoustics, AU-15, 2, Jun 1967.

Lanczos, C., Applied Analysis, Prentice-Hall, Inc., Englewood Cliffs, N.J., 1964.

Larson, A. G., and R. C. Singleton, "Real-Time Spectral Analysis on a Small General-Purpose Computer," AFIPS Fall Joint Computer Conf., Anaheim, Calif., 14-16 Nov 1967, to be published in vol. 31 proceedings.

Maling, G. C., Jr., W. T. Morrey, and W. W. Lang, "Digital Determination of Third-Octave and Full Octave Spectra of Acoustical Noise," IEEE Trans. Audio and Electroacoustics, AU-15, 2, Jun 1967.

Mason, J. J., and H. J. Zimmerman, Electronic Circuits, Signals, and Systems, John Wiley & Sons, New York, 1960.

Perini, J., "Side-Lobe Reduction Beam Shifting," IEEE Trans. Antennas and Propagation, AP-12, 6, Nov 1964, pp. 791-792.

Phister, M., Jr., Logical Design of Digital Computers, John Wiley & Sons, New York, 1958.

Singleton, R. C., "A Method for Computing the Fast Fourier Transform with Auxiliary Memory and Limited High-Speed Storage," IEEE Trans. Audio and Electroacoustics, AU-15, 2, Jun 1967.

Singleton, R. C., "On Computing the Fast Fourier Transform," Research Memorandum, Mathematical Sciences Department, Stanford Research Institute, Menlo Park, Calif., Jul 1967.

Singleton, R. C., and T. C. Poulter, "Spectral Analysis of the Call of the Male Killer Whale," IEEE Trans. Audio and Electroacoustics, AU-15, 2, Jun 1967.

Sloane, E. A., "An Introduction to Time Series Analysis," Time/Data Corporation, Palo Alto, Calif., Mar 1967.

Weaver, C. S., P. E. Mantey, R. W. Lawrence, and C. A. Cole, "Digital Spectrum Analyzers," Report SU-SEL-66-059 (TR 1809/1810-1), Stanford Electronics Laboratories, Stanford, Calif., Jun 1966.

Welch, L. R., "Computation of Finite Fourier Series," JPL Space Programs Summary No. 37-37, IV, Jet Propulsion Laboratory, Pasadena, Calif., Jan 1966.

Welch, P. D., "The Use of the Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging over Short, Modified Periodograms," IEEE Trans. Audio and Electroacoustics, AU-15, 2, Jun 1967.